university of groningen

State space and graphical models for estimating networks dynamics

PhD thesis

to obtain the degree of PhD at the
University of Groningen
on the authority of the
Rector Magnificus, Prof. E. Sterken
and in accordance with
the decision by the College of Deans.

This thesis will be defended in public on

Monday 17 february 2014 at 14:30 hrs.

by

Anani Lotsi

born on 14 november 1972
in Accra, Ghana

Supervisor:
Prof. E.C. Wit

Assessment committee:
Prof. E.R. van den Heuvel
Prof. M. van de Wiel
Dr. V. Vinciotti

ISBN: 978-90-367-6711-8

Contents

Chapter 1

Introduction

Genomic systems have become complex. Networks reconstruction are seen as an attractive paradigm of genomic science. The thesis is concerned with estimating networks dynamics. We have proposed two models: graphical mixture models (GMM) and state space models (SSM) and suggested two novel methods of inferences namely penalized Gaussian graphical mixture models (PGGMM) and penalized state space models (PSSM). Given the incompleteness nature of information, we propose the Expectation-Maximization (EM) algorithm as solution to such incomplete data problems. The thesis describes in detail two of the most popular applications of EM algorithm: estimating Gaussian graphical mixture models and estimating State Space Models. In this introductory chapter we give an overview of these models as well as the methods of estimation.

1.1 Graphical Model (GM)

Graphical models bring together graph theory and probability theory in a powerful formalism for multivariate statistical modeling. They are tools for formulating statistical models and algorithms for computing basic statistical quantities such as likelihoods and marginal probabilities.

In this thesis, a graph $\mathbb{G} = (V, E)$ is formed by a collection of (i) a finite set of vertices V, where $V = (1, 2, ..., p)$ represents the nodes in the graph and (ii) a set of edges E corresponding to (conditional) dependencies, where $E \subseteq V \times V$ is a subset of ordered pairs of distinct vertices (i, j). An edge is undirected if $(i, j) \in E \Rightarrow (j, i) \in E$. An ordered pair $(j, i) \in E$ denotes a directed edge from node i to node j and i is said to be a parent of j denoted $\mathbf{Pa}_j$. A graph is undirected if all edges are undirected as shown in Figure 1.1(a). A directed graph contains only directed edges, see Figure 1.1(b).

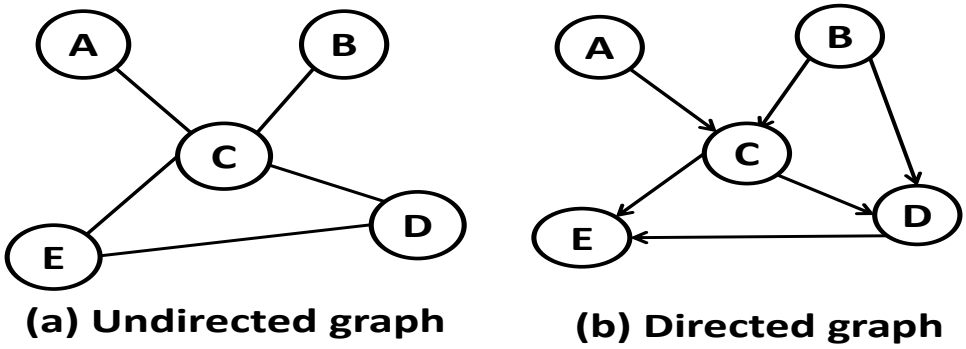

Figure 1.1. Two main kinds of graphical model, where (A, B, C, D) is interpreted as a vector of random variables.

In a graphical model, the vertices of a graph, i.e the set V corresponds to a collection of random variables

$$\mathbf{X} = (X_1, X_2, ..., X_p) \sim \mathbb{P},$$

where $\mathbb{P}$ is the probability distribution of $\mathbf{X}$. The pair $(\mathbb{G}, \mathbb{P})$ is referred to as a graphical model. Graphical models represent the relationships between a set of random variables through their joint distribution. They consist of a collection of probability distributions that factorize according to the structure of an underlying graph.

The study of graphical models has attracted a lot of attention in fields such as communication theory, control theory and bioinformatics; see for instance the books by Lauritzen (1996), Pearl (2000) and Whittaker (2009). Corresponding to directed and undirected graphs, two important classes of GM are directed and undirected GMs. Figure (1.1) depicts two main kinds of graphical models where nodes correspond to random variables and edges represent statistical dependencies between the variables. It is possible, though less common, to use a mixed directed and undirected representation (see, for example, the work on chain graphs by Lauritzen and Wermuth (1989) and Buntine (1995)).

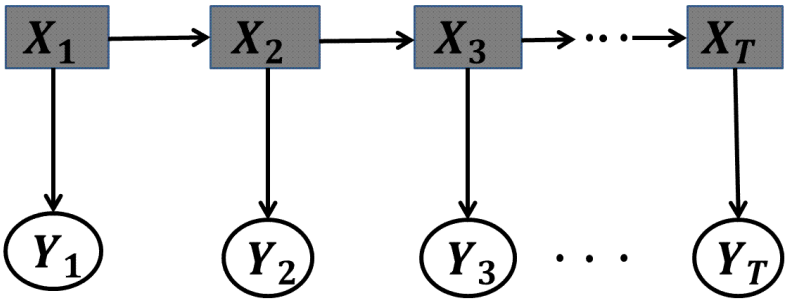

Figure 1.2. Linear dynamic system or hidden Markov models where $\mathbf{Y}_i$ is interpreted as vector of r.v representing gene expression levels at time t_i. The hidden factors are represented by $\mathbf{X}_i$

1.1.1 Directed graphical models

In a directed graphical model, an arc from A to B can be informally interpreted as indicating that A causes B. Each edge is directed from the parent node to the child node. A directed acyclic graph is a directed graph with no directed cycles. It is formed by a collection of vertices and directed edges, each edge connecting one vertex to another, such that there is no way to start at some vertex i and follow a sequence of edges that eventually loops back to i again.

A Bayesian network is a directed acyclic graph that encodes a joint probability distribution over a set of random variables $X_1, ..., X_p$.

Definition 1. *A Bayesian network is a pair $(\mathbb{G}, \mathbb{P}_{BN})$. The first component, $\mathbb{G}$, is a directed acyclic graph whose vertices correspond to the random variables $X_1, ..., X_p$, and whose edges represent direct dependencies between the variables. The graph $\mathbb{G}$ encodes independence assumptions: each variable X_1 is independent of its non-descendants given its parents in $\mathbb{G}$. The second component of the pair, namely $\mathbb{P}_{BN}$, factorizes over $\mathbb{G}$ and is given by*

$$\mathbb{P}_{BN}(x_1, ..., x_p) = \prod_{i=1}^{p} P(x_i | \mathbf{Pa}_{x_i}) \tag{1.1}$$

Directed graphical models are also called Bayesian networks, (Jensen, 1946). Dynamic Bayesian networks, (Fahrmeir and Kunstler, 2009) an extension of Bayesian networks to the analysis of time course data, explicitly account for time dependen-

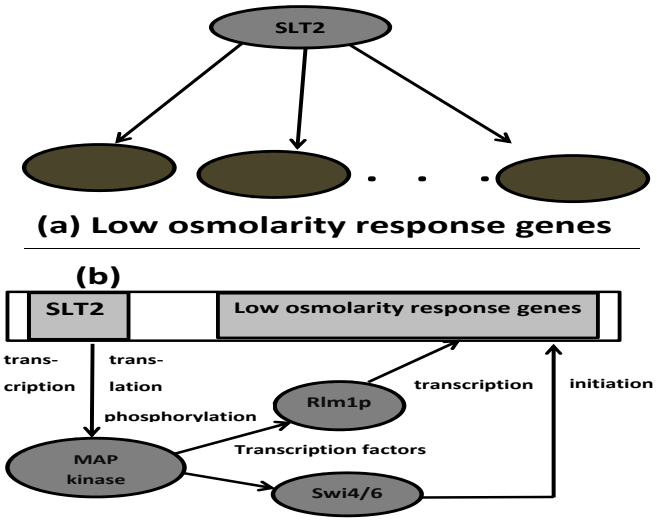

Figure 1.3. (a). **A simple Bayesian subnetwork reported in Peer et al. (2001) showing SLT2 as the parent gene responsible for the transcription of several low osmolarity response genes. (b) A Bayesian network where SLT2 through the enzyme MAP kinase triggers the translation process by activating two transcription factors. These transcription factors in turn initiate the transcription process by activating the expression level of low osmolarity response genes.**

cies. Dynamic Bayesian networks are often restricted to linear systems, state space model (SSM). Figure (1.2) depicts a typical example of a linear dynamic system.

The Bayesian network depicted in Figure 1.1 (b) corresponds to a factorization of the joint distribution function

$$P(A, B, C, D, E) = P(A)P(B)P(C|A, B)P(D|B, C)P(E|C, D),$$

whiles the linear dynamic system depicted in Figure (1.2) corresponds

$$P(X_{1,\ldots,T}, Y_{1,\ldots,T}) = P(X_1)P(Y_1|X_1)\prod_{t=2}^{T}\left[P(X_t|X_{t-1})P(Y_t|X_t)\right].$$

Figure 1.3(a) indicates another example of Bayesian subnetworks modelling reported in Peer et al. (2001) where several low osmolarity response genes are separated

by their parent gene, STL2. This indicates that gene SLT2 triggers positively the expression level of many low osmolarity response genes. In Figure 1.3(b), the gene SLT2 is transcribed and translated into two protein transcription factors Rlm1p and Swi4/6 through the enzyme MAP kinase. Both then activate the expression level of the low osmolarity response genes. The Bayesian networks can be used to explain the dependencies in the molecular system. Peer et al. (2001) used this framework to discover a finer structure of interactions between genes. Husmeier (2003) has showed how modelling with Bayesian networks can be used to assign novel putative functions to yet unannotated genes.

1.1.2 Undirected graphical models

The second common class of probabilistic graphical models is called a Markov network, which corresponds to undirected graphical model.

Definition 2. *(Whittaker, 2009) Let* $\mathbf{X} = (X_1, X_2, ..., X_p)$ *be a p-dimensional random vector and* $\mathbb{G}$ *a graph with nodes given by* I *where* $I = (i_1, ..., i_p)$*. An undirected graphical model for* $\mathbf{X}$ *is a family of probability distributions that satisfies the pairwise conditional independence restrictions inherent in* $\mathbb{G}$*, i.e,*

$$(i, j) \notin E \Leftrightarrow X_i \perp\!\!\!\perp X_j | X_{I\setminus\{i,j\}}.$$

In what follows we denote the set of neighbours of a node i with $ne(i)$, that is the set of $j \in V$ such that $(i, j) \in E$ and $(j, i) \in E$. The boundary of node i, $bd(i)$ will be defined as: $bd(i) = pa(i) \cup ne(i)$. The closure of node i is given by $cl(i) = \{i\} \cup bd(i)$. We also define separation as: two nodes, i and j are separated by a subset s if and only if all paths connecting the two pass through at least one member of the subset. This means all path from i to j intersect s.

Markov properties on undirected graph

Associated with an undirected graph $\mathbb{G} = (V, E)$ and a collection of random variables $(X_i)_{i\in V}$ we have a range of Markov properties. A probability measure $\mathbb{P}$ on $\mathbf{X}$ is said to obey:

(P) the pairwise Markov property, relative to $\mathbb{G}$, if for all non-adjacent vertices i and j,

$$X_i \perp\!\!\!\perp X_j | X_a \quad \text{where} \quad a = V \setminus \{i, j\}$$

Relating to example in Figure 1.1(a), we write $A \perp\!\!\!\perp E|(B, C, D)$

(L) the local Markov property, relative to $\mathbb{G}$, if for every vertex i, $a = bd(i)$ is its boundary set, and b the set of remaining vertices, i.e $b = V \setminus cl(i)$, then

$$X_i \perp\!\!\!\perp X_b | X_a$$

Relating to example in Figure 1.1(a), we write $A \perp\!\!\!\perp (B, D, E)|C$

(G) the global Markov property, relative to $\mathbb{G}$ if for all disjoint subsets (a, b, c) of V such that b and c are separated by a in the graph, then X_b and X_c are independent given X_a and we write

$$X_b \perp\!\!\!\perp X_c | X_a$$

Relating to example in Figure 1.1(a), if we suppose that (A, C, D) are disjoint subsets of V, we write $A \perp\!\!\!\perp D|C$

The relation between the Markov properties are described in proposition below.

Proposition 1. *For any undirected graph $\mathbb{G}$ and any probability distribution on* $\mathbf{X}$ *it holds that*

$$(G) \Rightarrow (L) \Rightarrow (P).$$

Proposition 2. *If it holds that for all disjoint subsets a, b, c and d that*

$$\text{if} \quad a \perp\!\!\!\perp b|(c \cup d) \quad \text{and} \quad a \perp\!\!\!\perp c|(b \cup d) \quad \Rightarrow \quad a \perp\!\!\!\perp (b \cup c)|d \tag{1.2}$$

then the Markov properties are all equivalent

Theorem 1.1.1. *Under the assumption (1.2), we have*

$$(G) \Leftrightarrow (L) \Leftrightarrow (P)$$

Proof. From proposition (1), we will show that $(P) \Rightarrow (G)$ by assuming that (a, b, s) are disjoints subsets and that s separates a from b in the graph G. Let the number of vertices in s be n. We will conduct the proof by backward induction on n. If $n = |V| - 2$, then both a and b have only one vertex and (P) holds.

Suppose $|s| = n < |V| - 2$ and assume that that $V = a \cup b \cup s$. This means that a or b has more than one element. Let suppose it is a. If $i \in a$ then $s \cup \{i\}$ separates $a \setminus \{i\}$ from b and also $s \cup a \setminus \{i\}$ separates $\{i\}$ from b. It follows by induction that:

$$a \setminus \{i\} \perp\!\!\!\perp b|s \cup \{i\} \quad \text{and} \quad i \perp\!\!\!\perp b|s \cup a \setminus \{i\}$$

and from (1.2), we have $a \perp\!\!\!\perp b|s$

Also if $a \cup b \cup s \subset V$, we can choose $i \in V \setminus (a \cup b \cup s)$, this implies that $s \cup \{i\}$ separates a and b, meaning $a \perp\!\!\!\perp b | s \cup \{i\}$. Further, we have either $b \cup s$ separates a from $\{i\}$ or $a \cup s$ separates b from $\{i\}$. In case we assume the first case, we then have $i \perp\!\!\!\perp a | b \cup s$ and from (1.2) it follows that $a \perp\!\!\!\perp b | s$ □

Furthermore, a probability distribution $\mathbb{P}$ on $\mathbf{X}$ is said to satisfy the (F) factorization property w.r.t graph $\mathbb{G}$, if $\mathbb{P}$ has density

$$f(x_1, ..., x_p) = \prod_{c \in C} \Psi_c(\mathbf{x}_c)$$

where C is a set of cliques of $\mathbb{G}$, i.e subsets of vertices which induce a complete subgraph but for which the addition of a further vertex renders the induced subgraph incomplete, $\Psi_c(\mathbf{x}_c)$ is a potential positive function of the variables $\mathbf{x}_c = \{x_i\}_{i \in C}$.

The undirected network depicted in Figure 1.1(a) corresponds to a factorization of the joint distribution function

$$P(A, B, C, D, E) = \Psi_{A,C}(A, C)\Psi_{B,C}(B, C)\Psi_{C,D,E}(C, D, E)$$

Proposition 3. *For any undirected graph $\mathbb{G}$ and any probability distribution on $\mathbf{X}$ it holds that*

$$(F) \Rightarrow (G) \Rightarrow (L) \Rightarrow (P)$$

Theorem 1.1.2. *(Hammersley and Clifford). A probability distribution with positive and continuous pdf w.r.t to a product measure μ satisfies the pairwise Markov property w.r.t to an undirected graph $\mathbb{G}$ if and only if it factorizes according to $\mathbb{G}$*

Proof. The proof requires the following lemma:

Lemma 1.1.3. *Consider a finite set V. let $\mathbb{M}$ and $\mathbb{N}$ be functions defined over all possible subsets of V. Then $\forall a \subseteq V$, the statement*

$$\mathbb{M}(a) = \sum_{b : b \subseteq a} \mathbb{N}(b) \tag{1.3}$$

is equivalent to the statement

$$\mathbb{N}(a) = \sum_{b : b \subseteq a} (-1)^{|a \setminus b|} \mathbb{M}(b) \tag{1.4}$$

where $|a|$ denotes the cardinality of the subset a.

Proof. We substitute Equation 1.4 into Equation 1.3 and show that Equation 1.3 follows.

$$\begin{aligned}
\sum_{b:b\subseteq a}\mathbb{N}(b) &= \sum_{b:b\subseteq a}\left[\sum_{c:c\subseteq b}(-1)^{|b\setminus c|}\mathbb{M}(c)\right] \\
&= \sum_{c:c\subseteq a}\sum_{b:c\subseteq b,b\subseteq a}\mathbb{M}(c)(-1)^{|b\setminus c|} \\
&= \sum_{c:c\subseteq a}\mathbb{M}(c)\sum_{b:c\subseteq b,b\subseteq a}(-1)^{|b\setminus c|} \\
&= \sum_{c:c\subseteq a}\mathbb{M}(c)\sum_{h:h\subseteq a\setminus c}(-1)^{|h|}
\end{aligned} \tag{1.5}$$

Note that

$$\sum_{h:h\subseteq a\setminus c}(-1)^{|h|}$$

is zero for all $a\setminus c$ except for the case when $a\setminus c=\emptyset$. Also, $a\setminus c=\emptyset$ only when $a=c$. This leads to

$$\sum_{c:c\subseteq a}\mathbb{M}(c)\sum_{h:h\subseteq a\setminus c}(-1)^{|h|}=\mathbb{M}(a)$$

□

Now to prove theorem (1.1.2), it suffices to prove that $(P)\Longrightarrow(F)$. Consider the joint density $f(x)$ where $x=\{x_\alpha\}$ such that x_α takes values in some space. For each element $a\subseteq X$, we define

$$H_a(x)=\ln f(x_a,x^*_{a^c})$$

where $(x_a,x^*_{a^c})$ has components x_α for $\alpha\in a$ and $x^*_{\alpha^c}$ for $\alpha\notin a$. Thus $H_a(x)$ depends on x only through x_a. Now define the following set of functions for all $a\subseteq X$.

$$n_a(x)=\sum_{b:b\subseteq a}(-1)^{|a\setminus b|}H_b(x) \tag{1.6}$$

This implies $n_a(x)$ also depends on x only through x_a. Using lemma (1.1.3), we obtain:

$$H_x(x)=\sum_{a:a\subseteq x}n_a(x) \tag{1.7}$$

Note also that $\ln f(x) = H_a(x)$. Defining $m_a(x_a) = \exp n_a(x_a)$ and taking exponential of both sides of (1.7), we obtain

$$\exp H(x) = \exp\left\{\sum_{a:a\subseteq x} n_a(x)\right\}$$

This implies

$$\exp\ln f(x) = \exp\left\{\sum_{a:a\subseteq x} n_a(x)\right\}$$

and finally

$$f(x) = \prod_{a:a\subseteq x} n_a(x)$$

It now remains to show that $n_a(x)$ vanishes unless the subset a is complete. To do that we make use of assumption (P). Let α, $\beta \in a$ such that there is no direct link between them. Let $c = a \setminus \{\alpha, \beta\}$. If we let $H_a = H_a(x)$. Then

$$n_a(x) = \sum_{b:b\subseteq c} (-1)^{|c\setminus b|} \left\{H_b - H_{b\cup\alpha} - H_{b\cup\beta} + H_{b\cup\{\alpha,\beta\}}\right\} \tag{1.8}$$

Define $d = x \setminus \{\alpha, \beta\}$, then by (P), we have $\alpha \perp\!\!\!\perp \beta | d$. Hence

$$\begin{aligned}
H_{b\cup\{\alpha,\beta\}} - H_{b\cup\alpha}(x) &= \ln \frac{f(x_b, x_\alpha, x_\beta, x^*_{d\setminus b})}{f(x_b, x_\alpha, x^*_\beta, x^*_{d\setminus b})} \\
&= \ln \frac{f(x_\alpha|x_b, x^*_{d\setminus b}) f(x_\beta, x_b, x^*_{d\setminus b})}{f(x_\alpha|x_b, x^*_{d\setminus b}) f(x^*_\beta, x_b, x^*_{d\setminus b})} \\
&= \ln \frac{f(x^*_\alpha|x_b, x^*_{d\setminus b}) f(x_\beta, x_b, x^*_{d\setminus b})}{f(x^*_\alpha|x_b, x^*_{d\setminus b}) f(x_\beta, x_b, x^*_{d\setminus b})} \\
&= \ln \frac{f(x_b, x^*_\alpha, x_\beta, x^*_{d\setminus b})}{f(x_b, x_\alpha, x^*_\beta, x^*_{d\setminus b})} \\
&= H_{b\cup\beta} - H_b(x)
\end{aligned} \tag{1.9}$$

From (1.8), $n_a(x)$ vanishes whenever there is no direct link between α and β, that is, $n_a(x)$ vanishes unless a is a complete set. □

Undirected graphical models are useful in modelling a variety of phenomena where one cannot naturally ascribe a directionality to the interactions between variables. A special member of the family of UGM includes the undirected Gaussian graphical model (GGM).

1.1.3 Gaussian Graphical Models (GGM)

In this section we consider the case that $\mathbf{X} = (X_1, X_2, ..., X_p)'$ is a continuous random vector modelled by a multivariate normal or Gaussian distribution.

Definition 3. *Multivariate Gaussian Distribution. A random vector* $\mathbf{X} = (X_1, X_2, ..., X_p)'$ *of continuous variables has a* p *variate multivariate normal distribution* $N_p(\mu, \Sigma)$ *with* $p \times 1$ *mean vector* μ *and* $p \times p$ *variance − covariance matrix* Σ *containing the entries* $\sigma_{ij} = Cov(X_i, X_j)$

$$\mu = \begin{pmatrix} \mu_1 \\ \vdots \\ \mu_p \end{pmatrix} \qquad and \qquad \Sigma = \begin{pmatrix} \sigma_{11} & \cdots & \sigma_{1p} \\ \vdots & & \vdots \\ \sigma_{p1} & \cdots & \sigma_{pp} \end{pmatrix}$$

where we assume that Σ *has full rank (this implies that it is positive definite), if the joint pdf has the form*

$$f(\mathbf{x}) = |2\pi\Sigma^{-1}| \exp\left\{-(\mathbf{x} - \mu)'\Sigma^{-1}(\mathbf{x} - \mu)/2\right\}.$$

In the above, μ and Σ are called the moment parameters. Some of the following theory is however better expressed through what are called the canonical parameters: $\Theta := \Sigma^{-1}$ is called the precision or concentration matrix, and $\beta = \Sigma^{-1}\mu$. We can easily see that the pdf in terms of these canonical parameters is given by

$$f(\mathbf{x}) = \exp\left\{\alpha + \beta'\mathbf{x} - \mathbf{x}'\Theta\mathbf{x}/2\right\}$$

where α is a normalizing constant. The above shows that the multivariate normal distributions form an exponential family. Writing out the matrix notation we see

$$f(\mathbf{x}) = \exp\left\{\alpha + \sum \beta_i x_i - \sum\sum \theta_{ij} x_i x_j/2\right\}$$

from where we see, using the factorization property, that:

$$\theta_{ij} = 0 \quad \text{if and only if} \quad x_i \perp\!\!\!\perp x_j | \mathbf{x}_{V\setminus\{i,j\}}.$$

Definition 4. *Gaussian Graphical Model (GGM). The* p *variate multivariate normal distribution* $N_p(0, \Sigma)$ *is graphical with respect to an undirected graph* $\mathbb{G} = (V, E)$ *if the corresponding entries in the precision matrix,* $\theta_{ij} = 0$ *whenever* $\{i, j\}$ *and* $\{j, i\} \notin E$.

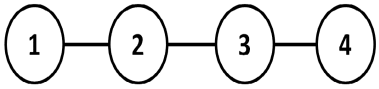

Figure 1.4. Graph of independence, where $(1, 2, 3, 4)$ denotes nodes.

The graph $\mathbb{G}$ represents the model where Θ, the concentration matrix is a positive definite matrix with $\theta_{ij} = 0$, whenever there are no edges between nodes i and j in $\mathbb{G}$. In undirected Gaussian graphical model, a missing edge implies conditional independence, and thus the problem of estimating a GGM is equivalent to estimating an inverse covariance matrix. For example, consider a four dimensional vector $\mathbf{X}$ as shown in Figure (1.4). A GGM for $\mathbf{X}$ is given by the inverse covariance matrix Θ of the form

$$\Theta = \begin{pmatrix} \theta_{11} & \theta_{12} & 0 & 0 \\ \theta_{12} & \theta_{22} & \theta_{23} & 0 \\ 0 & \theta_{23} & \theta_{33} & \theta_{34} \\ 0 & 0 & \theta_{34} & \theta_{44} \end{pmatrix},$$

where the remaining θ_{ij} are arbitrary, restrained only to ensure the matrix is symmetric and positive definite.

The corresponding Markov properties are listed as follows:

(P) $X_1 \perp\!\!\!\perp X_3|(X_2, X_4), \quad X_1 \perp\!\!\!\perp X_4|(X_2, X_3), \quad X_2 \perp\!\!\!\perp X_4|(X_1, X_3);$

(L) $X_1 \perp\!\!\!\perp X_3|X_2, \quad X_1 \perp\!\!\!\perp X_4|(X_2, X_3), \quad X_2 \perp\!\!\!\perp X_4|X_3;$

(G) $X_1 \perp\!\!\!\perp X_3|X_2, \quad X_2 \perp\!\!\!\perp X_4|X_3.$

Figure(1.5) depicts an example of subnetworks of Arabidopsis thaliana clock genes. It indicates no direct binding between genes CCA1 and TOC1. The two genes interact through some hidden mechanism (latent variable), as we will describe in chapter 3 and 4. In graphical model term, genes CCA1 and TOC1 are conditionally independent given the latent variable and the local Markov property can be translated as $CCA1 \perp\!\!\!\perp TOC1|$hidden variable. Also the graph indicates no direct interaction between genes LHY and CCA1. However the two genes interact with each other through the latent variable and the pairwise Markov property can be written as
$LHY \perp\!\!\!\perp CCA1|$(latent variable).

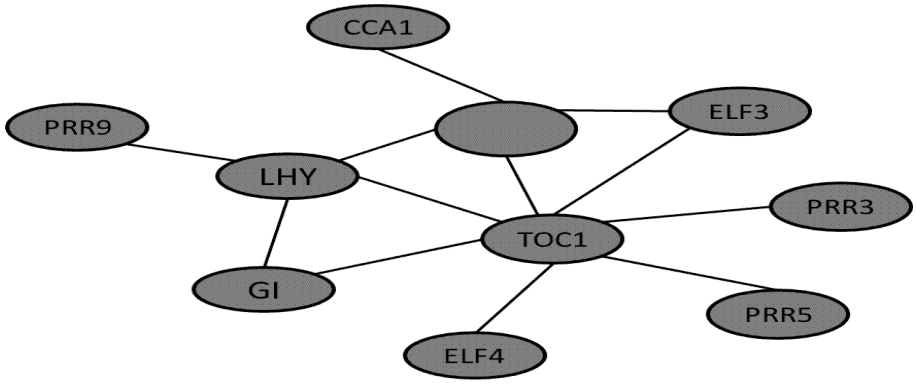

Figure 1.5. Example of undirected graph showing dependencies structure in the Arabidopsis thaliana clock genes.

1.1.4 Likelihood: Gaussian Graphical Models

Suppose we have a sample of independent copies $\mathbf{X}_1, ..., \mathbf{X}_N$ of $\mathbf{X}$, with $\mathbf{X}_i = (X_{i1}, ..., X_{ip})^{'}$ and $\mathbf{x}_i = (x_{i1}, ..., x_{ip})^{'}$. Then, the log-likelihood in terms of (Θ, μ) is given by

$$l(\Theta, \mu) = \frac{N}{2} \ln |\Theta| - \frac{N}{2} tr(\Theta S) - \frac{N}{2} (\bar{\mathbf{x}} - \mu)^{'} \Theta (\bar{\mathbf{x}} - \mu),$$

where $S = \frac{1}{N} \sum (x_i - \bar{\mathbf{x}})(x_i - \bar{\mathbf{x}})^{'}$. For fixed Θ this is maximized by $\hat{\mu} = \bar{\mathbf{x}}$ which makes the last term equal to zero. We are then left with the profile likelihood,

$$l(\Theta) = \frac{N}{2} \ln |\Theta| - \frac{N}{2} tr(\Theta S),$$

where $tr(\Theta S) = \sum_{i,j} s_{ij} \theta_{i,j}$. We then conclude that the only elements $s_{i,j}$ of S for which $\theta_{i,j} \neq 0$ will contribute to the likelihood. This leads to the following:

Theorem 1.1.4 (Estimation for Gaussian Graphical Models)**.** *Given* $\mathbf{X}_i$ *and* $\mathbf{x}_i$ *above, it can be shown using exponential family theory that*

1. *S and $\bar{\mathbf{x}}$ are sufficient statistics for Θ and μ*

2. *For complete graph, the ML-estimator of Σ is $\hat{\Sigma} = S$, the ML-estimator for μ in any graph is $\hat{\mu} = \bar{\mathbf{x}}$*

Graphical model in systems biology

In systems biology graphical models are employed to describe and to identify interdependencies among genes and gene products, with the aim to better understand the molecular mechanisms of the cell. GMs are promising tools for the analysis of gene interactions because they allow the stochastic description of networks association and dependency structures in complex highly structured data. They are perfectly suited for modelling biological process in the cell such as biochemical interactions and regulatory activities. As a result many graphical models, such as Bayesian networks (Friedman, 2004), vector-autoregressive (VAR) models (Fujita et al., 2007), state space models (Husmeier, 2003; Rangel et al., 2004) have already been applied to genomic data and put to use in expression analysis.

Although graphical models are promising for the analysis of gene interaction, a major practical problem encountered in their application in systems biology is the high dimensionality of the data with respect to the sample size. As a result, it is not trivial to apply Gaussian graphical models to high dimensional genomic data for the parameters describing the GM quickly outnumber the data points. The reason being that inferring a large scale precision matrix from relatively few data is an ill-posed problem that requires care. Motivated by these challenges, some efforts are now being undertaken to avoid the dimensionality problems stated above. These approaches include dimension reduction prior to classic Gaussian graphical models analysis (Kishino and Waddell, 2000; Xintao et al., 2003), limited order partial correlations (de la Fuente et al., 2004; Wille and Peter, 2006), and regularization techniques (Nicolai et al., 2006). In subsequent chapters of this thesis, we apply an L_1 regularization technique to graphical models and state space models in order to infer large networks.

1.1.5 State Space Model

State space models (SSM), (Fahrmeir and Kunstler, 2009) are examples of directed graphical models. They relate observations $\{y_t\}_{t=1,2,\ldots,T.}$ on a response variable Y to unobserved "states" $\{x_t\}_{t=1,2,\ldots,T}$ by an observation model for y_t given x_t. The model assumes that the observation at time t was generated by some process whose state x_t is hidden from the observer. Second it assumes that the state of this hidden process satisfies the Markov property: that is given the value of x_{t-1}, the current state x_t is independent of all the states prior to $t-1$. The output also satisfies a Markov property with respect to the states: given x_t, y_t is independent of the states and the observations at all other time indices.

From Equation (1.1), the joint distribution of the sequence of states and obser-

vations can be factored in the following way:

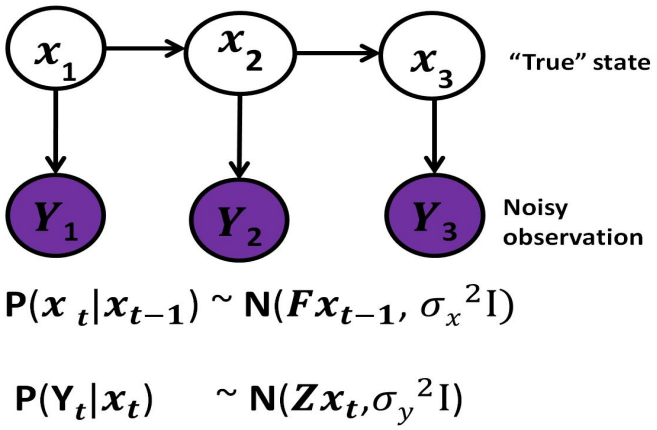

Figure 1.6. Linear State-space model (SSM). The r.v. x_t represent the true unobserved states. They are assumed to be Gaussian distribution with mean Fx_{t-1} and variance-covariance $\sigma_x^2 I$. Y_t are the observation variables supposed to be Gaussian distribution with mean Zx_t and variance-covariance $\sigma_y^2 I$.

$$P\left(x_{1:T}, y_{1:T}\right) = P\left(x_1\right) P\left(y_1|x_1\right) \prod_{t=2}^{T} P\left(x_t|x_{t-1}\right) P\left(y_t|x_t\right) \tag{1.10}$$

The state transition probability $P\left(x_t|x_{t-1}\right)$ can be decomposed into deterministic and stochastic components:

$$x_t = g_t\left(x_{t-1}\right) + \eta_t,$$

where g_t is the deterministic transition function determining the mean of x_t given x_{t-1}. In a similar manner, the observation (y_t) can be decomposed as

$$y_t = f_t\left(x_t\right) + \xi_t,$$

where η_t and ξ_t are state and observation noise vectors respectively. In the case where the transition and output functions are linear with i.i.d. sequences $\eta_t \sim N\left(0, \sigma_x^2 I\right)$, $\xi_t \sim N\left(0, \sigma_y^2 I\right)$ with an independent initial value $x_0 \sim N\left(a_0, \sigma_{x_0}^2\right)$, the model is called a linear Gaussian state-space model or linear dynamical systems (LDS) (Roweis and Ghahramani, 1999), also known as a Kalman filter model (Brown and Hwang, 1997):

$$x_t = Fx_{t-1} + \eta_t \tag{1.11}$$

and

$$y_t = Zx_t + \xi_t \tag{1.12}$$

where F and Z represent the transition matrices and the design matrices respectively. The graphical representation of the model is depicted in Figure (1.6).

A straightforward and powerful extension of this model is to allow the dynamics and observation models to include feedback from previous data points:

$$\begin{cases} x_t = Fx_{t-1} + Ay_{t-1} + \eta_t, & \text{where } \eta_t \sim N(0, \sigma_x^2) \\ y_t = Zx_t + By_{t-1} + \xi_t, & \text{where } \xi_t \sim N(0, \sigma_y^2) \\ x_0 = 0, & y_0 = 0 \end{cases} \tag{1.13}$$

where A and B are the $(k \times p)$ input-to-state and $(p \times p)$ input-to-observation matrices respectively. The dimension of the state variable x and the data y are k and p respectively. We will consider this model thoroughly in chapter 3 and 4.

1.1.6 Connection between GGMs and SSMs

Let $y_O \sim N(0, \Theta_O)$ where Θ_O denotes the marginal precision matrix of the observed variables. In addition, we consider the setting in which the hidden variables x_H and the observed variables y_O are jointly Gaussian with covariance matrix Σ. The question we address here is as follows: How is the precision matrix $\Theta = \Sigma^{-1}$ connected to the SSM parameters? In another words, we want to recover the network of the observed variables while still taking into account the uncertainty about the hidden components.

Given our input dependent SSM above, and considering 2 different time points, the dependencies between the variables are explained in table below

		Variables			
		y_1	x_1	y_2	x_2
variables	y_1	ξ_1	Z	B	A
	x_1	Z	η_1	0	F
variables	y_2	B	0	ξ_2	Z
	x_2	A	F	Z	η_2

From the model, we can infer the following:

$$x_1 \sim N(0, \sigma_x^2 I)$$

$$y_1|x_1 \sim N(Zx_1, \sigma_y^2 I_p)$$

and marginally

$$y_1 \sim N(0, \underbrace{Z\sigma_x^2 Z^{'} + \sigma_y^2 I_p)}_{\Sigma_{y_1}}$$

Also

$$cov(x_1, x_2) = F\sigma_x^2$$
$$cov(x_1, y_1) = Z\sigma_x^2$$

Next,

$$x_2|(x_1, y_1) \sim N(Fx_1 + Ay_1, \sigma_x^2 I)$$

and marginally

$$x_2 \sim N\left(0, \underbrace{F^2 x_\sigma^2 + A[Z\sigma_x^2 Z^{'} + \sigma_y^2 I_p]A^{'} + \sigma_x^2 + 2FZ\sigma_x^2 A^{'}}_{\Sigma_{x_2}}\right)$$

Futhermore

$$y_2|(x_2, y_1) \sim N(Zx_2 + By_1, \sigma_y^2 I_p)$$

and marginally

$$y_2 \sim N\left(0, \underbrace{Z\Sigma_{x_2} Z^{'} + B\Sigma_{y_1} B^{'} + 2Z(FZ^{'}\sigma_x^2 + A\sigma_y^2)B^{'} + \sigma_y^2 I_p}_{\Sigma_{y_2}}\right)$$

Also

$$cov(y_1, y_2) = B\Sigma_{y_1}$$
$$cov(y_2, x_1) = (\underbrace{ZF + AZ + BZ}_{C})\sigma_x^2$$

We can now write that

$$\begin{bmatrix} y_1 \\ y_2 \\ x_1 \\ x_2 \end{bmatrix} \sim N\left(0, \underbrace{\begin{pmatrix} \Sigma_{y_1} & B\Sigma_{y_1} & Z\sigma_x^2 & A\Sigma_{y_1} \\ (B\Sigma_{y_1})^{'} & \Sigma_{y_2} & C\sigma_x^2 & Z\Sigma_{x_2} \\ (Z\sigma_x^2)^{'} & (C\sigma_x^2)^{'} & \sigma_x^2 & F\sigma_x^2 \\ A(\Sigma_{y_1})^{'} & (Z\Sigma_{x_2})^{'} & (F\sigma_x^2)^{'} & \Sigma_{x_2} \end{pmatrix}}_{\Sigma}\right)$$

Derivation of the precision matrix

We write down the correlation matrices among the observed data y_t and between the observed data y_t and the hidden states x_t.

$$cor(y_t, y_s|rest) = \begin{cases} 0, & \text{if } |s-t| > 1 \\ B, & \text{if } |s-t| = 1 \\ I, & \text{if } |s-t| = 0 \end{cases}$$

Next

$$cor(y_t, x_s|rest) = \begin{cases} 0, & \text{if } |s-t| > 1 \\ \frac{\sigma_y}{\sigma_x} A', & \text{if } (s-t) = 1 \\ Z\frac{\sigma_x}{\sigma_y}, & \text{if } (s-t) = 0 \\ 0, & \text{if } (s-t) = -1 \end{cases}$$

Furthermore,

$$cor(x_t, x_s|rest) = \begin{cases} 0, & \text{if } |s-t| > 1 \\ F, & \text{if } |s-t| = 1 \\ I, & \text{if } |s-t| = 0 \end{cases}$$

The precision matrix of the conditional statistics of the observed variables y_O given the latent variables is given by Θ_O which is simply a submatrix of the full precision matrix Θ. We now have

$$\begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ x_1 \\ x_2 \\ \vdots \end{bmatrix} \sim N\left(0, \underbrace{\begin{pmatrix} \Theta_O & \Theta_{OH} \\ \Theta_{HO} & \Theta_H \end{pmatrix}}_{\Theta}\right)$$

where the structure of Θ is of the form,

	Obs. var.				**Hidden var.**				
	y_1	y_2	y_3	$\cdots$	x_1	x_2	x_3	x_4	$\cdots$
y_1	I_{p*p}	B	0	$\cdots$	$Z\frac{\sigma_x}{\sigma_y}$	$\frac{\sigma_y}{\sigma_x}A^{'}$	0	0	
y_2		I	B	$\ddots$	0	$Z\frac{\sigma_x}{\sigma_y}$	$\frac{\sigma_y}{\sigma_x}A^{'}$	0	$\cdots$
$\vdots$				$\ddots$					
x_1					I	F	0		
x_2						I	F	0	
$\vdots$							$\ddots$		$\ddots$

The marginal statistics of the observed variables y_O are given by the marginal precision matrix $\hat{\Theta}_O$, which according to Schur complement (Horn and Johnson, 1990), is given by

$$\hat{\Theta}_O = \Theta_O - \Theta_{OH}\Theta_H^{-1}\Theta_{HO} \tag{1.14}$$

where Θ_O, Θ_{OH}, Θ_H are the corresponding submatrices of the full precision matrix. The matrix Θ_O specifies the concentration matrix of the conditional statistics of the observed variables given the hidden components. It is usually sparse as compared to the matrix $\Theta_{OH}\Theta_H^{-1}\Theta_{HO}$; the latter is of low-rank. The reason being that the conditional statistics are given by a sparse graphical model whiles the number of hidden components is smaller than the number of observed variables. Equation (1.14) can be viewed as a decomposition of the $\hat{\Theta}_O$ into a sparse and a low-rank components. From above for simplicity, suppose $F = 0$ and let $c = \frac{\sigma_y}{\sigma_x}$, we then have

$$\Theta_{OH}\Theta_H^{-1}\Theta_{HO} = \begin{bmatrix} (\frac{ZZ^{'}}{c^2} + c^2A^{'}A) & A^{'}Z^{'} & \cdots & 0 \\ ZA & (\frac{ZZ^{'}}{c^2} + c^2A^{'}A) & A^{'}Z^{'}\cdots & 0 \\ \ddots & \ddots & \ddots & \\ & & & \ddots \\ & & ZA & (\frac{ZZ^{'}}{c^2} + c^2A^{'}A) \end{bmatrix} \tag{1.15}$$

and from Equation 1.14, we have

$$\hat{\Theta}_O = \begin{bmatrix} (I - \frac{ZZ'}{c^2} - c^2 A'A) & B - A'Z' & \cdots & 0 \\ B' - ZA & (I - \frac{ZZ'}{c^2} - c^2 A'A) & B - A'Z' \cdots & 0 \\ \ddots & \ddots & \ddots & \\ & & & \ddots \\ & B' - ZA & (I - \frac{ZZ'}{c^2} - c^2 A'A) & \end{bmatrix} \quad (1.16)$$

1.1.7 Identifiability issues of SSMs

There is a fundamental problem for system identification using SSMs. By identifiability, we mean a unique parametrization exists. Parameters of a model are not identifiable if there exists infinite number of parametrization that yield the same likelihood. In this case the statistical problem of estimating the parameters is ill-posed. If we simply estimate parameters of the SSMs without any constraints on the parameter space, it lacks identifiability. We explain here three important properties in SSMs that relate to identifiability.

stability

Recall the state Equation (1.11), $x_t = Fx_{t-1} + \eta_t$, recursively, we can show that

$$x_t = F^n x_{t-n} + \eta_t^*$$

If v is any eigen vector, then there exists $Fv = \lambda v$ where λ is an eigen value. Suppose $v_1, ..., v_n$ are basis of the eigen vectors such that

$$x_{t-1} = a_1 v_1 + ... + a_n v_n$$

then

$$\begin{aligned} Fx_{t-1} &= \sum_{i=1}^{n} a_i F v_i \\ &= \sum_{i=1}^{n} a_i \lambda_i v_i \end{aligned}$$

where $\lambda_i < 1$ indicates shrinking and $\lambda_i > 1$ shows expansion in the direction of v_i. Therefore the model (1.11) and (1.12) will be stable if the matrix F has spectra radius less than one. In other words, we require the eigen values of F to be less than one in magnitude.

controlability

Our SSM is said to be controllable if it can evolve from any arbitrary initial state, say x_0 to any desirable state, x_k in a finite time period. Mathematically, we can write the state equation (1.11) as

$$x_k = F^k x_0 + F^{k-1}\eta_1 + F^{k-2}\eta_2 + ... + I\eta_k$$

for any finite time k. This implies

$$x_k - F^k x_0 = \left[F^{k-1}, F^{k-2}, ..., I\right] \begin{bmatrix} \eta_1 \\ \eta_2 \\ \vdots \\ \eta_k \end{bmatrix}$$

Therefore for any x_k and x_0 our system will be controllable provided we can choose judiciously inputs $\eta_1, ...\eta_k$ such that we can move from x_0 to x_k. For that matter we require the matrix $\left[F^{k-1}, F^{k-2}, ..., I\right]$ to be of full rank.

observability

A SSM is said to be observable if, when the noise vectors were to be 0, every initial state, x_0 can be determined from observable sequence of y_k over a finite number of sampling period. This condition is satisfied if and only if the inverse of the observability matrix is available. That is the matrix

$$\left[ZF^{k-1}, \quad \dots \quad , ZF, \quad Z\right]$$

is of full rank.

In Chapter 3 and 4 we will apply SSM in order to infer network and further constraints were imposed on the model to overcome the poor identifiability of the SSM.

1.2 The Expectation-Maximization algorithm

The main method of inference used in this thesis is the Expectation- Maximization (EM) algorithm. Frequentist estimation by and large relies on maximum likelihood (ML) estimators. It consists of maximizing the likelihood across the parameter space. Special cases of the EM algorithm were developed before it was formally introduced by Dempster et al. (1977). The EM algorithm has become a popular method of

inference in statistical estimation problems involving incomplete data, i.e, data with some missing or latent or hidden observations or problems that can be posed in a similar form, such as mixture models. Subsequent chapters will demonstrate how the EM algorithm can be used in Graphical models involving mixture distributions as well as in SSM .

The EM algorithm is an iterative tool to compute the maximum likelihood estimate in data characterized by the presence of missing, or hidden or latent observations. This optimization can be difficult especially if the data consist of missing or latent parts. The intuition behind ML is to estimate the parameter(s) for which the observed sample is most likely. It possesses some optimality properties as discussed in George and Berger (1996). Each iteration of the EM algorithm consists of an expectation step (E-step) followed by a maximization step (M-step). In the E-step, the hidden variables are "estimated" as conditional expectations given the observed data and current estimates of the model parameters. In our SSM, the Kalman filtering algorithm is precisely the E-step. The later is achieved by computing the conditional expectation of the (log) likelihood of the "complete" data. The M-step maximizes the complete likelihood function across the parameter space given the estimate of the missing data from the E-step.

Let $\mathbf{y}$ be a random vector which results from a parametrized family. The EM algorithm aims at finding a value of Ω which maximizes $P(\mathbf{y}|\Omega)$. In most cases, especially when differentiation is to be used, it is easier to work with the natural logarithm, known as the log likelihood and denoted by $l(\Omega)$, defined as,

$$l(\Omega) = \ln P(\mathbf{y}|\Omega). \tag{1.17}$$

A ML estimate is the same regardless of whether we maximize the likelihood or the log-likelihood function, since the logarithm is a monotone increasing transformation. The EM algorithm is an iterative procedure for maximizing $l(\Omega)$. Assume that after the k^{th} iteration the current estimate for Ω is given by Ω_k. Since the goal is to maximize $l(\Omega)$, we wish to compute an updated estimate Ω such that:

$$l(\Omega) > l(\Omega_k). \tag{1.18}$$

Stated differently, we may want to maximize the difference:

$$l(\Omega) - l(\Omega_k) = \ln P(\mathbf{y}|\Omega) - \ln P(\mathbf{y}|\Omega_k). \tag{1.19}$$

So far, we have not considered any missing or hidden variables. However in many practical problems, such variables can be naturally constructed and the EM algorithm offers a reliable framework for their inclusion. The idea behind the EM

algorithm stems from the fact we observe data $\mathbf{y}$; but if we had observed $(\mathbf{y}, \mathbf{x})$ with $\mathbf{x}$ as the missing data, then estimation of parameter vector Ω through maximum likelihood would have been much easier.

The likelihood $P(\mathbf{y}|\Omega)$ may be written in terms of the hidden variables $\mathbf{x}$ as,

$$P(\mathbf{y}|\Omega) = \int_{\mathbf{x}} P(\mathbf{y}|\mathbf{x}, \Omega) P(\mathbf{x}|\Omega) \, d\mathbf{x}. \tag{1.20}$$

Therefore we may rewrite Equation (1.19) as,

$$\begin{aligned}
l(\Omega) - l(\Omega_k) &= \ln \left(\int_{\mathbf{x}} P(\mathbf{y}|\mathbf{x}, \Omega) P(\mathbf{x}|\Omega) \right) d\mathbf{x} - \ln P(\mathbf{y}|\Omega_k) \\
&= \ln \left(\int_{\mathbf{x}} P(\mathbf{y}|\mathbf{x}, \Omega) P(\mathbf{x}|\Omega) . \frac{P(\mathbf{x}|\mathbf{y}, \Omega_k)}{P(\mathbf{x}|\mathbf{y}, \Omega_k)} \right) d\mathbf{x} - \ln P(\mathbf{y}|\Omega_k) \\
&= \ln \left(\int_{\mathbf{x}} P(\mathbf{x}|\mathbf{y}, \Omega_k) . \frac{P(\mathbf{y}|\mathbf{x}, \Omega) P(\mathbf{x}|\Omega)}{P(\mathbf{x}|\mathbf{y}, \Omega_k)} \right) d\mathbf{x} - \ln P(\mathbf{y}|\Omega_k) \\
&\geq \int_{\mathbf{x}} P(\mathbf{x}|\mathbf{y}, \Omega_k) \ln \left(\frac{P(\mathbf{y}|\mathbf{x}, \Omega) P(\mathbf{x}|\Omega)}{P(\mathbf{x}|\mathbf{y}, \Omega_k)} \right) d\mathbf{x} - \ln P(\mathbf{y}|\Omega_k) \\
&= \int_{\mathbf{x}} P(\mathbf{x}|\mathbf{y}, \Omega_k) \ln \left(\frac{P(\mathbf{y}|\mathbf{x}, \Omega) P(\mathbf{x}|\Omega)}{P(\mathbf{x}|\mathbf{y}, \Omega_k)} \right) d\mathbf{x} \\
&- \int_{\mathbf{x}} P(\mathbf{x}|\mathbf{y}, \Omega_k) \ln P(\mathbf{y}|\Omega_k) d\mathbf{x} \\
&= \int_{\mathbf{x}} P(\mathbf{x}|\mathbf{y}, \Omega_k) \ln \left(\frac{P(\mathbf{y}|\mathbf{x}, \Omega) P(\mathbf{x}|\Omega)}{P(\mathbf{x}|\mathbf{y}, \Omega_k) P(\mathbf{y}|\Omega_k)} \right) d\mathbf{x} \\
&:= \delta(\Omega|\Omega_k)
\end{aligned} \tag{1.21}$$

We can now write that

$$l(\Omega) \geq l(\Omega_k) + \delta(\Omega|\Omega_k) := L(\Omega|\Omega_k). \tag{1.22}$$

The function $L(\Omega|\Omega_k)$ is bounded from above by the likelihood function $l(\Omega)$.

Next,

$$\begin{aligned} L\left(\Omega_k|\Omega_k\right) &= l\left(\Omega_k\right) + \delta\left(\Omega_k|\Omega_k\right) \\ &= l\left(\Omega_k\right) + \int_{\mathbf{x}} P\left(\mathbf{x}|\mathbf{y},\Omega_k\right) \ln\left(\frac{P\left(\mathbf{y}|\mathbf{x},\Omega_k\right) P\left(\mathbf{x}|\Omega_k\right)}{P\left(\mathbf{x}|\mathbf{y},\Omega_k\right) P(\mathbf{y}|\Omega_k)}\right) d\mathbf{x} \\ &= l\left(\Omega_k\right) + \int_{\mathbf{x}} P\left(\mathbf{x}|\mathbf{y},\Omega_k\right) \ln\frac{P\left(\mathbf{y},\mathbf{x}|\Omega_k\right)}{P\left(\mathbf{y},\mathbf{x}|\Omega_k\right)} d\mathbf{x} \\ &= l\left(\Omega_k\right) \end{aligned} \tag{1.23}$$

Any Ω which maximizes $L\left(\Omega|\Omega_k\right)$ will also maximize $l(\Omega)$ and in order to attain the highest increment in the value of $l(\Omega)$, the EM algorithm calls for selecting Ω such that $L\left(\Omega|\Omega_k\right)$ is maximized. Let this updated value be Ω_{k+1}. Then

$$\begin{aligned} \Omega_{k+1} &= \text{argmax}_{\Omega}\left\{L\left(\Omega|\Omega_k\right)\right\} \\ &= \text{argmax}_{\Omega}\left\{l\left(\Omega_k\right) + \delta\left(\Omega|\Omega_k\right)\right\} \\ &= \text{argmax}_{\Omega}\left\{l\left(\Omega_k\right) + \int_{\mathbf{x}} P\left(\mathbf{x}|\mathbf{y},\Omega_k\right) \ln\left(\frac{P\left(\mathbf{y}|\mathbf{x},\Omega\right) P\left(\mathbf{x}|\Omega\right)}{P\left(\mathbf{x}|\mathbf{y},\Omega_k\right) P(\mathbf{y}|\Omega_k)}\right) d\mathbf{x}\right\} \\ &= \text{argmax}_{\Omega}\left\{\int_{\mathbf{x}} P\left(\mathbf{x}|\mathbf{y},\Omega_k\right) \ln P\left(\mathbf{y}|x,\Omega\right) P\left(x|\Omega\right) d\mathbf{x}\right\} \\ &= \text{argmax}_{\Omega}\left\{\int_{\mathbf{x}} P\left(\mathbf{x}|\mathbf{y},\Omega_k\right) \ln\frac{P\left(\mathbf{y},\mathbf{x},\Omega\right)}{P\left(\mathbf{x},\Omega\right)}\frac{P\left(\mathbf{x},\Omega\right)}{P\left(\Omega\right)} d\mathbf{x}\right\} \\ &= \text{argmax}_{\Omega}\left\{\int_{\mathbf{x}} P\left(\mathbf{x}|\mathbf{y},\Omega_k\right) \ln P\left(\mathbf{y},\mathbf{x}|\Omega\right) d\mathbf{x}\right\} \\ &= \text{argmax}_{\Omega}\left\{E_{\mathbf{x}|\mathbf{y},\Omega_k}\left[\ln P\left(\mathbf{y},\mathbf{x}|\Omega\right)\right]\right\} \end{aligned} \tag{1.24}$$

From above, the two steps are now apparent and each iteration of the EM algorithm involves the following:

1. E-step: "Estimate" the complete likelihood by computing the conditional expectation,
$$E_{\mathbf{x}|\mathbf{y},\Omega_k}\left[\ln P\left(\mathbf{y},\mathbf{x}|\Omega\right)\right].$$

2. M-step: Maximize the expected complete likelihood with respect to Ω to obtain the next estimates.

It is well known that the log-likelihood calculated with the $(k+1)^{th}$ iterative estimated parameters is larger than that of the k^{th} iterative estimated parameters. This is the ascent property of the EM algorithm giving reason why the EM-algorithm works in general. It is also necessary to point out that the gain in maximizing $L(\Omega|\Omega_k)$ instead of $l(\Omega)$ stems from the fact that the maximization of $L(\Omega|\Omega_k)$ is tractable and easier as compared to a direct maximization of $l(\Omega)$.

The procedure to obtain the maximum likelihood estimator of the parameter vector Ω is summarized below:

1. Select initial values of $\hat{\Omega}_0$ that is, start with initial guess for the parameters $\hat{\Omega}_0$

2. At the k^{th} step, calculate the conditional expectation of the log likelihood in Equation (1.24) (E-step)

3. Determine the next iterative estimated parameters $(\hat{\Omega}_{k+1})$ that maximizes conditional expectation of the log likelihood. (M-step) and compute the corresponding log likelihood.

4. Iterate step 2 and 3 until the log likelihood $(l(\hat{\Omega}|\mathbf{Y}))$ is converged

1.2.1 The Kalman-Filtering Algorithm

Preliminaries

The Kalman filter has been considered as one of the optimal solutions to many data prediction, filtering and smoothing problems. In this context, it is used to estimate the hidden or latent states in the E-step of the EM algorithm in chapter 3 and 4. Here we describe the basic concepts that one needs to know to design and implement a Kalman filter, and later introduce the Kalman filter algorithm. In the model description Equations (1.11) and (1.12) we assume that the random variables ξ_t and η_t that represent the measurements and process noise respectively are independent and distributed according to the following:

$$\xi_t \sim N(0, \sigma_y^2 I_p) \tag{1.25}$$

$$\eta_t \sim N(0, \sigma_x^2 I) \tag{1.26}$$

$$\eta_0 \sim N(0, \sigma^2_{x_0}) \tag{1.27}$$

It is possible to rewrite our model Equations (1.11) and (1.12) in just one equation called Final Form which, will be more convenient for easy derivations of moments of states and observations. To do that, one needs to realize that we can write Equation (1.11) recursively as

$$x_1 = Fx_0 + \eta_1$$
$$x_2 = F^2x_0 + F\eta_1 + \eta_2$$

and then we obtain

$$x_t = F^t x_0 + \sum_{i=0}^{t-1} F^i \eta_{t-i}$$

and substituting this into Equation (1.12) we obtain

$$y_t = Z\left\{F^t x_0 + \sum_{i=0}^{t-1} F^i \eta_{t-i}\right\} + \xi_t$$

where the superscript F^t indicates F raised to the power t. We shall refer to the above equation as the Final form From which, the mean of the states x_t and of the observations y_t are obtained directly as follows:

$$\begin{aligned} E(x_t) &= F^t E(x_0) \\ &= F^t a_0, \end{aligned} \tag{1.28}$$

and

$$\begin{aligned} E(y_t) &= ZF^t E(x_0) \\ &= ZF^t a_0. \end{aligned} \tag{1.29}$$

Next if we let Π_t denote the unconditional covariances of the state vector, then

$$\Pi_t = E\left[\left(x_t - E(x_t)\right)\left(x_t - E(x_t)\right)'\right],$$

from which it follows that the covariances of y is

$$E\left[\left(y_t - E(y_t)\right)\left(y_t - E(y_t)\right)'\right] = Z\Pi_t Z' + \sigma_y^2 I_p.$$

Rewriting the model

The goal is to find the distribution of x_t, conditional on y_t. The SSM consists of a hidden system x_t with initial probability density $p(x_0)$ which evolves over time as an indirect or partially observed first order Markov process according to the conditional probability density $p(x_t|x_{t-1})$ whose distribution is:

$$x_t|x_{t-1} \sim N(Fx_{t-1}, \sigma_x^2 I) \tag{1.30}$$

The observations y_t are conditionally independent given the state and are generated according to the conditional probability density $p(y_t|x_t)$ with distribution given by:

$$y_t|x_t \sim N(Zx_t, \sigma_y^2 I_p). \tag{1.31}$$

One also need to realize that: $\{x_t\}$ is Markovian, i.e.

$$p(x_t|x_{t-1}, \eta_1, ..., \eta_{t-1}) = p(x_t|x_{t-1}), \tag{1.32}$$

$$p(x_t|x_{t-1}, y_1, ..., y_{t-1}) = p(x_t|x_{t-1}). \tag{1.33}$$

Next from from Bayes rule, it follows that

$$p(x_{t-1}|x_t) = p(x_t|x_{t-1})\frac{p(x_{t-1})}{p(x_t)}. \tag{1.34}$$

Recall that as

$$x_t = F^t x_0 + \sum_{i=0}^{t-1} F^i \eta_{t-i},$$

it follows also that

$$x_t \sim N(F^t a_0, \Pi_t). \tag{1.35}$$

Note that the joint distribution of (y_t, x_t) is

$$\begin{pmatrix} x_t \\ y_t \end{pmatrix} \sim N\left(\begin{bmatrix} F^t a_0 \\ ZF^t a_0 \end{bmatrix}, \begin{bmatrix} \Pi_t & \Pi_t Z \\ (\Pi_t Z)' & Z\Pi_t Z' + \sigma_y^2 I_p \end{bmatrix}\right).$$

The Kalman filter equations (KFE)

The Kalman filter equations (KFE) is a set of equations that provides an efficient computational means to calculate the conditional expectation of the states of a process, given the observations or measurements. Recall that the objective is to find

$$p(x_t|Y_s) \tag{1.36}$$

where $Y_s = \{y_1, y_2, ..., y_T\}$, representing all available information up to time T. It can also be viewed as the history of responses for an individual up to and including time T. This gives rise to three different cases depending on the range of values of s. That is, if $s < t$ we talk of prediction, filtering if $s = t$ and finally $s > t$ refers to smoothing. Here we discuss only the first two cases, i.e prediction and filtering.

The prediction equations

We begin by finding the density $p(x_t|Y_{t-1})$. The following lemma is also important to the derivation of the prediction equations.

Lemma 1.2.1. *Given that x and y are any random variable and if $x \sim N(\upsilon_x, S_x)$ and $y|x \sim N(Tx, S_y)$ with T an invertible square matrix then:*

$$\int_{-\infty}^{\infty} p(y|x)p(x)dx \sim N(T\upsilon_x, TS_xT^{'} + S_y) \tag{1.37}$$

We also introduce the following notations

$$\tilde{x}_t = E(x_t|Y_{t-1}) \tag{1.38}$$

where $\tilde{x}_t$ denotes the predicted or prior state estimate variable at time t given knowledge of the process prior to time t.

$$\begin{aligned} \tilde{P}_t &= E\left[(x_t - \tilde{x}_t)(x_t - \tilde{x}_t)^{'}|Y_{t-1}\right] \\ &= var(x_t|Y_{t-1}) \end{aligned} \tag{1.39}$$

therefore $\tilde{P}_t$ is the corresponding predicted or prior state estimate error covariance.

The filtered parameters $\hat{x}_t$ and $\hat{P}_t$ are also defined by:

$$\hat{x}_t = E(x_t|Y_t) \tag{1.40}$$

and

$$\begin{aligned}\hat{P}_t &= E\left[(x_t - \tilde{x}_t)(x_t - \tilde{x}_t)'|Y_t\right] \\ &= var(x_t|Y_{t-1}) \end{aligned} \tag{1.41}$$

Now the predictive conditional density $p(x_t|Y_{t-1})$ can be calculated as follows:

$$\begin{aligned}p(x_t|Y_{t-1}) &= \int_{-\infty}^{\infty} p(x_t, x_{t-1}|Y_{t-1})dx_{t-1} \\ &= \int_{-\infty}^{\infty} p(x_t|x_{t-1})p(x_{t-1}|Y_{t-1})dx_{t-1} \end{aligned} \tag{1.42}$$

where $p(x_t|x_{t-1})$ was calculated from the model and was given in Equation (1.30); and $p(x_{t-1}|Y_{t-1})$ represents the previous solution to the filtering problem. Equation (1.42) represents the predictive density step from which we realize that the recursion starts from $p(x_{t-1}|Y_{t-1})$. It is important to realize the following:

$$x_0|Y_0 \sim N(a_0, P_0) \tag{1.43}$$

and it follows that

$$x_{t-1}|Y_{t-1} \sim N(\hat{x}_{t-1}, \hat{P}_{t-1}) \tag{1.44}$$

Applying the lemma to Equation (1.42), the predictive step density is distributed according to:

$$x_t|Y_{t-1} \sim N\left(F\hat{x}_{t-1}, \left[F\hat{P}_{t-1}F' + \sigma_x^2 I\right]\right) \tag{1.45}$$

In combination with our notations in Equations (1.38) and (1.39), the predictive step equations give:

$$\tilde{x}_t = F\hat{x}_{t-1} \tag{1.46}$$

$$\tilde{P}_t = F\hat{P}_{t-1}F' + \sigma_x^2 I \tag{1.47}$$

Clearly, the predictive step equations is predicting x_t using $\tilde{x}_t$ only on the basis of $\hat{x}_{t-1}$. In the first predictive step, we can say that $\tilde{x}_1$ predicts $\hat{x}_1$ on the basis of $\hat{x}_0$. Therefore as a by-product of the predictive step, we quickly obtain the predictive density $p(y_t|Y_{t-1})$; however, we also need to realize that conditional on $\{x_t\}$, current observations y_t are independent of past states $x_{t-1}, x_{t-2}, ..., x_0$, i.e.

$$p(y_t|x_t, x_{t-1}, ..., x_0) = p(y_t|x_t).$$

Now

$$\begin{aligned} p(y_t|Y_{t-1}) &= \int_{-\infty}^{\infty} p(y_t|x_t, Y_{t-1})p(x_t|Y_{t-1})dx_t \\ &= \int_{-\infty}^{\infty} p(y_t|x_t)p(x_t|Y_{t-1})dx_t \end{aligned} \tag{1.48}$$

where $p(y_t|x_t)$ is known from the model and was given in Equation (1.31) and $p(x_t|Y_{t-1})$ comes from previous prediction step and $x_t|Y_{t-1}$ is distributed according

$$x_t|Y_{t-1} \sim N(\tilde{x}_t, \tilde{P}_t) \tag{1.49}$$

Therefore combining Equations (1.31) and (1.49) coupled with the lemma, the observation prediction density $p(y_t|Y_{t-1})$ step has $y_t|Y_{t-1}$ whose distribution is given by:

$$y_t|Y_{t-1} \sim N\left(Z\tilde{x}_t, \left[Z\tilde{P}_t Z^{'} + \sigma_y^2 I_p\right]\right) \tag{1.50}$$

If we let

$$\tilde{y}_t = E(y_t|Y_{t-1}) \tag{1.51}$$

$$v_t = y_t - \tilde{y}_t \tag{1.52}$$

where v_t is the measurement innovation or the residual and reflects the discrepancy between the predicted measurement $\tilde{y}_t$ and the actual observation y_t. Then the observation prediction equation step becomes:

$$\tilde{y}_t = Z\tilde{x}_t \tag{1.53}$$

$$\Sigma_t = Z\tilde{P}_t Z^{'} + \sigma_y^2 I_p \tag{1.54}$$

where Σ_t represent the observation prediction covariance.

It can be seen that the predictor equations are responsible for projecting forward the current state and error covariance estimates to obtain prior estimates for the next time.

The filtered equations step

Filtering means estimating the current state, given responses up to the present. Now from $t-1$ we want to project into t and ultimately finding an equation that computes a posterior state estimate $\hat{x}_t$ probably as a linear combination of the predictive state estimate or prior estimate $\hat{x}_t$ and a weighted residual v_t. This process is referring to as filtering step or update step. We begin by finding the filtered density $p(x_t|Y_t)$

$$\begin{aligned} p(x_t|Y_t) &= p(x_t|Y_{t-1}, y_t) \\ &= \frac{p(x_t, y_t|Y_{t-1})}{p(y_t|Y_{t-1})} \\ &= \frac{p(y_t|x_t, Y_{t-1})p(x_t|Y_{t-1})}{p(y_t|Y_{t-1})} \\ &= \frac{p(y_t|x_t)p(x_t|Y_{t-1})}{p(y_t|Y_{t-1})} \end{aligned} \tag{1.55}$$

This is the update or filtered density step and it is important to realize that $p(y_t|x_t)$ is determined by the model and was given in Equation (1.31), the second factor $p(x_t|Y_{t-1})$ comes from the prediction density step and was given in Equation (1.45) and the last factor $p(y_t|Y_{t-1})$ is the observation prediction density and was given in Equation (1.50). The filtered density has the interpretation that the filtered distribution $p(x_{t-1}|Y_{t-1})$ is propagated forwards by the dynamics for one time step to reveal a new "prior" distribution at time t. This distribution is then modulated by the observation y_t, incorporating the new evidence into the filtered distribution; this is also referred to as predictor-corrector method.

Now substituting Equations (1.31), (1.45) and (1.50) into Equation (1.55) gives:

$$\begin{aligned} p(x_t|Y_t) &= C \exp -\frac{1}{2}[(y_t - Zx_t)' R^{-1}(y_t - Zx_t) \\ &+ (x_t - \tilde{x}_t)' \tilde{P}_t^{-1}(x_t - \tilde{x}_t) - (y_t - \tilde{y}_t)' \Sigma_t^{-1}(y_t - \tilde{y}_t)] \end{aligned} \tag{1.56}$$

where C is a constant, $R = \sigma_y^2 I_p$. Re-arranging Equation (1.56) gives

$$p(x_t|Y_t) = C \exp \left\{ -\frac{1}{2}(x_t - \hat{x}_t)' \hat{P}_t^{-1}(x_t - \hat{x}_t) \right\} \tag{1.57}$$

where $\hat{x}_t$ and $\hat{P}_t$ represent posterior state estimate at step t given the observation y_t or filtered state variables, and posterior estimate error covariance respectively and are given by:

$$\hat{x}_t = \tilde{x}_t + K_t v_t \tag{1.58}$$

$$\hat{P}_t = \tilde{P}_t - K_t \Sigma_t K_t^{'} \tag{1.59}$$

and

$$K_t = \tilde{P}_t Z^{'} \Sigma_t^{-1} \tag{1.60}$$

is the Kalman gain and is chosen to be the gain or blending factor that minimizes the posterior error covariance in Equation (1.59). We provide a full derivation of the Kalman gain K using an alternative Minimization of Mean Square Error (MMSE) approach.

Derivation of Kalman gain

The filter can be constructed as a mean square error minimizer and for the optimal filter, it must be possible to correctly model the system errors using Gaussian distributions. Let the covariances of the two noises models be

$$Q = E(\eta_t \eta_t^{'}) \tag{1.61}$$

and

$$R = E(\xi_t \xi_t^{'}) \tag{1.62}$$

Given the model described in Equations (1.11) and (1.12) let

$$\hat{P}_t = MSE = E(e_t e_t^{'})$$

where

$$e_t = x_t - \hat{x_t}$$

then $\hat{P}_t$ becomes

$$\hat{P}_t = E\left[(x_t - \hat{x_t})(x_t - \hat{x_t})^{'}\right]$$

Given that $\tilde{x_t}$ is the prior estimate of x; then

$$\hat{x_t} = \tilde{x_t} + K_t(y_t - Z\tilde{x_t})$$

therefore $\hat{P}_t$ becomes

$$
\begin{aligned}
\hat{P}_t &= E[(x_t - \tilde{x_t} - K_t(y_t - Z\tilde{x_t}))(x_t - \tilde{x_t} - K_t(y_t - Z\tilde{x_t}))^{'}] \\
&= E[(x_t - \tilde{x_t} - K_t Z x_t - K_t\xi_t + K_t Z\tilde{x_t}) \\
&\times (x_t - \tilde{x_t} - K_t Z x_t - K_t\xi_t + K_t Z\tilde{x_t})^{'}] \\
&= E[((x_t - \tilde{x_t}) - K_t Z(x_t - \tilde{x_t}) - K_t\xi_t) \\
&\times ((x_t - \tilde{x_t}) - K_t Z(x_t - \tilde{x_t}) - K_t\xi_t)^{'}] \\
&= E[((I - K_t Z)(x_t - \tilde{x_t}) - K_t\xi_t)((I - K_t Z) \\
&\times (x_t - \tilde{x_t}) - K_t\xi_t)^{'}] \\
&= (I - K_t Z)\underbrace{E\left[(x_t - \tilde{x_t})(x_t - \tilde{x_t})^{'}\right]}_{\tilde{P}_t}(I - K_t Z)^{'} + K_t\underbrace{E\left[\xi\xi_t^{'}\right]}_{R}K_t^{'} \\
&= (I - K_t Z)\tilde{P_t^{'}}(I - K_t Z)^{'} + K_t R K_t^{'} \qquad (1.63)
\end{aligned}
$$

Now the MSE error may be minimized by minimizing the trace of $\hat{P}_t$. The trace of $\hat{P}_t$ is first differentiated with respect to K_t and the result set to zero to find the condition of this minimum. That is

$$\frac{d(Tr[\hat{P}_t])}{d(K_t)} = 0$$

Expanding Equation (1.63) gives

$$\hat{P}_t = \tilde{P}_t - K_t Z\tilde{P}_t - \tilde{P}_t Z K_t^{'} + K_t(Z\tilde{P}_t Z^{'} + R)K_t^{'} \qquad (1.64)$$

and taking the trace gives

$$Tr(\hat{P}_t) = Tr(\tilde{P}_t) - 2Tr(K_t Z\tilde{P}_t) + Tr(K_t(Z\tilde{P}_t Z^{'} + R)K_t^{'}) \qquad (1.65)$$

differentiating with respect to K_t, we have

$$\frac{d(Tr[\hat{P}_t])}{d(K_t)} = -2(Z\tilde{P}_t)^{'} + 2K_t(Z\tilde{P}_t Z^{'} + R) = 0 \qquad (1.66)$$

This implies

$$(Z\tilde{P}_t)^{'} = K_t(Z\tilde{P}_t Z^{'} + R) \qquad (1.67)$$

and

$$
\begin{aligned}
K_t &= \tilde{P}_t Z^{'}(Z\tilde{P}_t Z^{'} + R)^{-1} \\
&= \tilde{P}_t Z^{'}\Sigma_t^{-1} \qquad (1.68)
\end{aligned}
$$

It can be shown that the Hessian of $Tr(\hat{P}_t)$ w.r.t K_t is positive semi-definite and thus the Kalman gain in (1.68) is indeed a minimum. Therefore we write that the filtered density or the posterior distribution $p(x_t|Y_t)$ is distributed according to

$$\begin{aligned} x_t|Y_t &\sim N(E[x_t], E[(x-\hat{x}_t)(x-\hat{x}_t)^{'}] \\ &\sim N(\hat{x}_t, \hat{P}_t) \end{aligned} \tag{1.69}$$

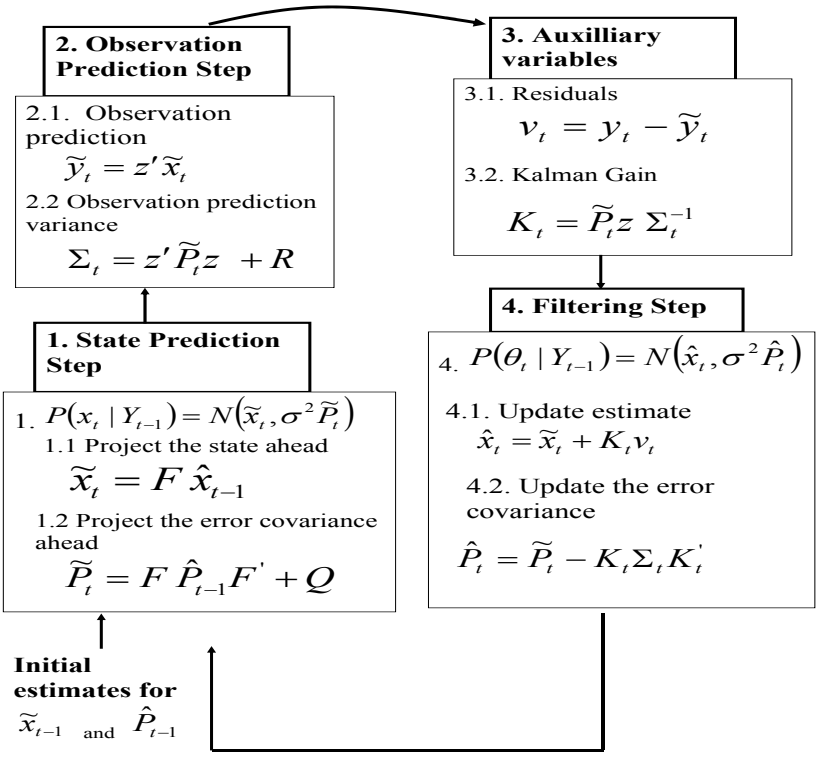

Figure 1.7. Kalman Filter Algorithm

The filtered or the update equations are responsible for the feedback; that is incorporating a new observation into prior estimate to obtain an improved posterior estimate. This completes the recursive filter. The recursive algorithm is summarized below.

Kalman filter algorithm

Given the data y_1, y_2,..., y_{t-1}, y_t
Step1: Initial estimates for $\hat{x}_{t-1}$ and $\hat{P}_{t-1}$
Step2: Prediction density or time update equations

$$x_t|Y_{t-1} \sim N\left(F\hat{x}_{t-1}, \left[F\hat{P}_{t-1}F' + Q\right]\right)$$

and projecting the state ahead

$$\tilde{x}_t = F_t\hat{x}_{t-1}$$

and projecting the error covariance ahead

$$\tilde{P}_t = F_t\hat{P}_{t-1}F_t^{'} + Q.$$

Step3: Observation and variance prediction

$$\tilde{y}_t = Z\tilde{x}_t$$

$$\Sigma_t = Z\tilde{P}_tZ^{'} + R.$$

Step4: Residuals and Kalman gain

$$v_t = y_t - \tilde{y}_t$$

$$K_t = \tilde{P}_tZ^{'}\Sigma_t^{-1}.$$

Step5: Filtered density or observation update equations

$$x_t|Y_t \sim N(\hat{x}_t, \hat{P}_t)$$

with update estimate

$$\hat{x}_t = \tilde{x}_t + K_tv_t$$

and update error covariance

$$\hat{P}_t = \tilde{P}_t - K_t\Sigma_tK_t^{'}.$$

We summarize the algorithm in Figure (1.7). The dynamics of the Kalman filter is indicated in Figure (1.8). The latter indicates that x_0 predicts x_1 using the predicted estimate $\tilde{x}_1$ without the observation y_1; but since we know y_1 the filter quickly update

$\tilde{x}_1$ with the help of y_1 to get the new posterior estimates $\hat{x}_1$. Hence after each time and measurement update pair, the process is repeated with the previous posterior estimates used to project or predict the new prior estimates. Therefore it is clear that recursively $\hat{x}_1$ is a function of the data y_1, $\hat{x}_2$ a function of the data y_1 and y_2 and ultimately $\hat{x}_n$ a function of the data y_1, y_2,..., y_n. This recursive nature defines a probabilistic generative model of how the system evolves over time and of how we observe this hidden state evolution; this can also be interpreted as a special case of the more general framework of Bayesian networks which we shall explore in subsequent chapters.

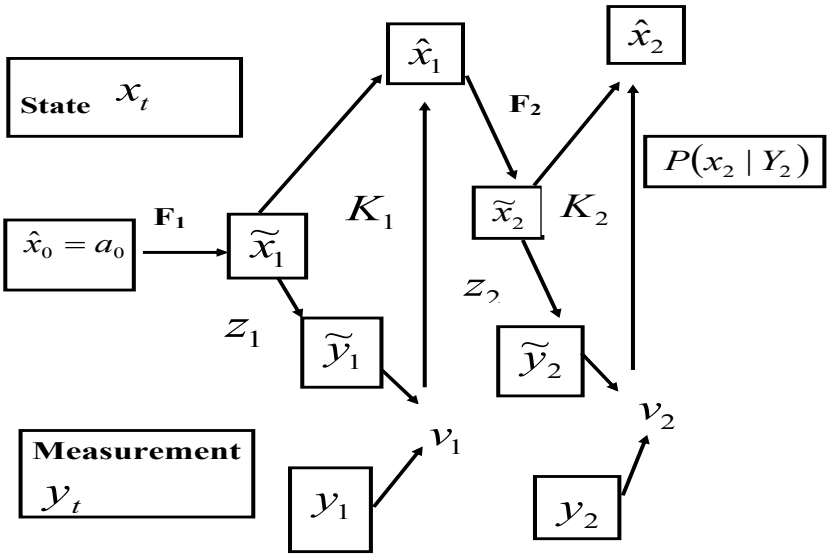

Figure 1.8. Kalman filter dynamics

The solution to the smoothing problem is slightly different as we now need to proceed backwards and evaluate the influence of future observation on our estimate of states in the past. The recursion is summarized below:

$$x_{t|T} = \hat{x}_{t|T} + K^*_{t|T}\left[x_{t+1|T} - \tilde{x}_{t+1|t}\right] \tag{1.70}$$

and

$$P_{t|T} = \hat{P}_t - K^*_t\left[P_{t+1|T} - P_{t+1|t}\right](K^*_t)' \tag{1.71}$$

where $\hat{x}_{t|T}$ and $P_{t|T}$ denote the smoothed values of x and corresponding variance covariance matrix respectively. Also K^*_t is the gain matrix playing a role similar to the Kalman gain and is computed backwards recursively.

In chapter 3 and 4 we implement the Kalman filter algorithm in the E-step of the EM algorithm. This enables us to calculate the hidden components from the model. The corresponding M-step is based on the LASSO/LARS regularization technique through an L_1 penalized inference.

1.2.2 The LASSO

The LASSO is an L_1 penalized regression technique introduced by Tibshirani (1996). It is a popular tool for sparse linear regression, especially for problems in which the number of variables p exceeds the number of observations. In Chapters (2) and (4) we make use of the LASSO technique in the M-step of the EM-algorithm as the inference method. We give a brief summary of the theory underlying LASSO.

Suppose we have a linear regression problem $\mathbf{Y} = \mathbf{Xb} + \mathbf{e}$, where $\mathbf{Y} \in R^n$ is the response, $\mathbf{X} \in R^{n\times p}$ is the design matrix, $\mathbf{b} \in R^p$ is the vector of unknown coefficients and $\mathbf{e} \in R^n$ is the noise vector. The goal is to find an estimate of the regression vector $\hat{\mathbf{b}}$ with good predictive performance and at the same time sparse.

Tibshirani (1996), introduced the concept of Least Absolute Shrinkage and Selection Operator known as LASSO. This method uses an L_1 norm constraint for regression estimation. Suppose for the moment that $p < n$ and $\mathbf{X}$ has full rank p. The unbiased ordinary least square (OLS) solution for our problem satisfies:

$$\arg\min_{\mathbf{b}}(||\mathbf{Y} - \mathbf{Xb}||_2),$$

where, $||.||_2$ refers to the sum of square elements of the vector, i.e. $||\mathbf{X}||_2 = \sum_{i=1}^n \mathbf{x}_i^2$. The solution is given by $\hat{\mathbf{b}} = \mathbf{b}_{OLS} = (\mathbf{X}'\mathbf{X})^{-1}\mathbf{X}'\mathbf{Y}$. In high dimensions, the OLS solution produces a relatively complex model in the sense that it usually becomes unstable. To overcome this problem, the LASSO constraints the solution, and its formulation is as follows:

$$\arg\min_{\mathbf{b}}(||\mathbf{Y} - \mathbf{Xb}||_2) \quad \text{subject to } ||\mathbf{b}||_1 \leq c \tag{1.72}$$

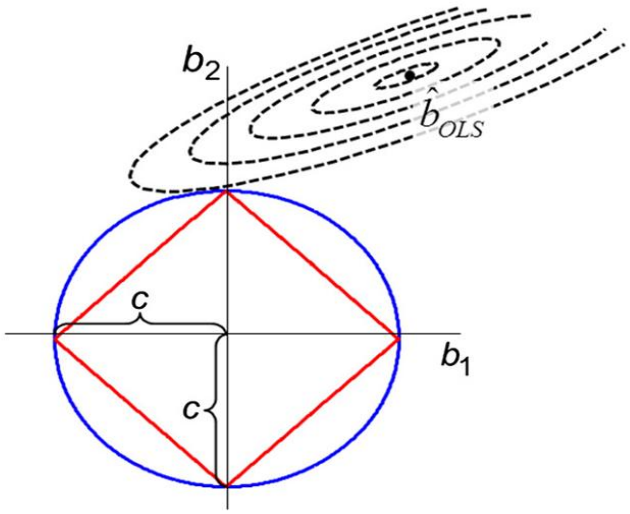

Figure 1.9. Two dimensional regression problem with estimates $\hat{b}_1$ and $\hat{b}_2$, $\hat{b}_{OLS}$ is the OLS estimates. The L_1 and the L_2 constraints are represented by an L_1 ball (rotated square) and L_2 ball (disc) respectively.

where c is the tuning parameter. For an appropriate bound c, this returns a sparse solution. Note that in Equation (1.72), as c increases the constraint $||\mathbf{b}||_1 \leq c$ relaxes and the solution gets closer to the OLS solution. However for small c such that $c << ||\mathbf{b}_{OLS}||_1$, there exist a unique solution and that the solution tends to be sparse. This is the essence of LASSO method.

To illustrate the LASSO method further, consider a two dimensional regression problems with regression coefficients represented by ($\mathbf{b} = [b_1, b_2]$). We want to compare the L_1 norm solution to the L_2 norm given by

$$\arg\min_{\mathbf{b}}(||\mathbf{Y} - \mathbf{X}\mathbf{b}||_2) \quad \text{subject to} \quad ||\mathbf{b}||_2 \leq c \tag{1.73}$$

The feasible solution for this problem is a disk. Figure (1.9) indicates the geometric representation of the parameter space where we show the L_2 norm constraint by the disc. From Figure (1.9), the optimal solution to Equation (1.73) occurs at points where the loss contour touches the feasible set of solution. Clearly this solution is not sparse.

However turning to L_1 norm constraint, the L_1 constraint is an area within the rotated square (L_1 ball) around the origin. The solution to this constrained optimization problem is the first point where the loss contour touches the rotated square. The L_1 ball has corners on the coordinates axes where at least one parameter, b_1,

is exactly zero. Hence the L_1 constraint always leads to some regression coefficients being exactly zero. Thus the LASSO solution is always sparse and enhances model interpretability.

1.2.3 Least Angle Regression "LARS"

We write LAR for least angle regression and LARS to include LAR as well as LASSO. We implement LARS by (Efron et al., 2004) or optimization with L_1-regularization constraint in chapter (4) in the M-step as an inference method. It turns out to be helpful and computationally feasible approach for finding sparse solutions in high dimension and by so rendering model interpretation easier. Here we give briefly the theory behind the LARS method.

Suppose we have available a large collection of possible covariates $\mathbf{X} = (\mathbf{X}_1, \mathbf{X}_2, ..., \mathbf{X}_p)$ with corresponding predictors $\mathbf{b} = (b_1, b_2, ..., b_p)$ from which we hope to select a parsimonious set for the efficient prediction of a response variable $\mathbf{Y}$. As before the LASSO chooses $\hat{\mathbf{b}}$ by minimizing $\arg\min_{\mathbf{b}}(||\mathbf{Y} - \mathbf{X}\mathbf{b}||_2^2)$ such that $\sum_{k=1}^{p} |\hat{b}_k| \leq c$.

The LARS algorithm exploits the special structure of the LASSO problem, and provides an efficient way to compute the solutions simultaneously for all values of the tuning parameter c. The LARS algorithm starts with $\hat{b}_k = 0 \quad \forall \quad k$, and find the predictor most correlated with the output variable, say $\mathbf{x}_j$. LARS takes the largest step possible in the direction of this predictor until some other predictor, say $\mathbf{x}_m$, has as much correlation with the current residual. Then LARS instead of continuing along the direction of $\mathbf{x}_j$, proceeds rather in a direction equiangular between the two predictors until another third covariate $\mathbf{x}_l$ enters its way into the "most correlated" set. LARS further proceeds in the direction equiangular between the variables $(\mathbf{x}_j, \mathbf{x}_m, \mathbf{x}_l)$, that is along the "least Angle direction" until the next predictor enters, and so on.

In essence, LARS builds up estimate $\hat{\mu} = \mathbf{X}\hat{\mathbf{b}}$ successively, each step one variable is added to the model. Starting with $\hat{\mu}_0$, with only 2 covariates and let $q(\hat{\mu})$ denote the vector of current correlations,

$$q(\hat{\mu}) = \mathbf{X}'(\mathbf{y} - \hat{\mu}).$$

From Figure (1.10), $(\bar{\mathbf{y}}_2 - \hat{\mu}_0)$ has a smaller angle with $\mathbf{x}_1$ than with $\mathbf{x}_2$, this implies $q_1(\hat{\mu}_0) > q_2(\hat{\mu}_0)$. LARS augments $\hat{\mu}_0$ in the direction of $\mathbf{x}_1$, that is

$$\hat{\mu}_1 = \hat{\mu}_0 + \hat{\gamma}_1 \mathbf{x}_1.$$

LARS chooses $\hat{\gamma}_1$ in such a way that $(\bar{\mathbf{y}}_2 - \hat{\mu}_1)$ bisects the angle between $\mathbf{x}_1$ and $\mathbf{x}_2$. The next LARS estimate is

$$\hat{\mu}_2 = \hat{\mu}_1 + \hat{\gamma}_2 u_2,$$

where $\hat{\gamma}_2$ is chosen to make $\hat{\mu}_2 = \bar{\mathrm{y}}_2$. Figure (1.10) explains the algorithm for $p = 2$ variables. The LARS procedure works as follows:

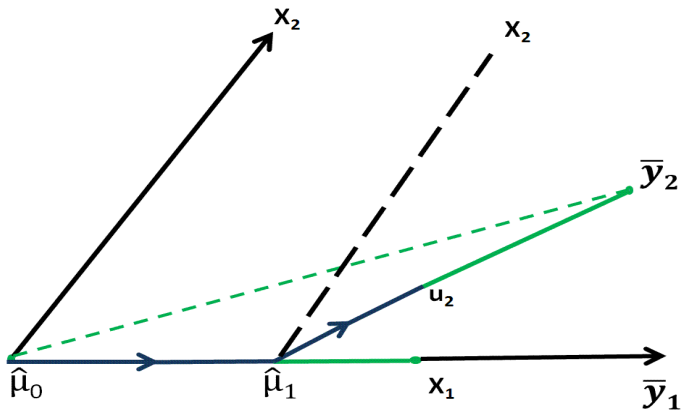

Figure 1.10. Geometrical representation of the LARS algorithm. $\bar{\mathrm{y}}_2$ is the projection of y into linear space spanned by x_1 and x_2. We start with $\hat{\mu}_0 = 0$. Given that $\bar{\mathrm{y}}_2 - \hat{\mu}_0$ has a smaller angle with x_1 than with x_2, LARS then augments $\hat{\mu}_0$ in the direction of x_1 i.e. $\hat{\mu}_1 = \hat{\mu}_0 + \hat{\gamma}_1 \mathrm{x}_1$. The choice of $\hat{\gamma}_1$ is critical here. LARS chooses $\hat{\gamma}_1$ such that $\bar{\mathrm{y}}_2 - \hat{\mu}_1$ bisects the angle between x_1 and x_2. The next LARS estimate now is $\hat{\mu}_2 = \hat{\mu}_1 + \hat{\gamma}_2 \mu_2$.

1. Start with all the coefficients in our model $\hat{b}_1, \hat{b}_2, ..., \hat{b}_p$ equal to zero.

2. Find the predictor $\mathbf{x}_j$ most correlated with $\mathbf{y}$ i.e making the smallest angle with our response $\mathbf{y}$.

3. Increase the coefficient b_j in the direction of the sign of its correlation with $\mathbf{y}$. Compute the residuals $r = \mathbf{y} - \hat{\mathbf{y}}$. Stop when some other predictor $\mathbf{x}_m$ has as much correlation with r as $\mathbf{x}_j$ has. At that point LARS switches to a direction that is "equiangularly" between the two predictors $\mathbf{x}_j$ and $\mathbf{x}_m$.

4. We continue in this way until a third variable $\mathbf{x}_l$ joins the"most correlated" set and we move "equiangularly" between $\mathbf{x}_j, \mathbf{x}_m, \mathbf{x}_l$. If we had p original predictors we continue in this way for $p-1$ steps and take as our last move a jump to the OLS fit using all p predictors.

5. suppose we have completed $k-1$ steps of the LARS algorithm, then the k^{th} step will see us introduce a new variable $\mathbf{x}_k$ say. It can be shown that the

LARS estimate for the k^{th} step $\hat{\mu}_k$ lies along the line between $\hat{\mu}_{k-1}$ and $\hat{\mathbf{y}}_k$, the OLS fit to the response using $\mathbf{x}_1, ..., \mathbf{x}_k$. See Figure (1.10) for a geometrical interpretation.

We summarize the LARS algorithm below:

1. Set $\hat{\mu}_0 = \mathbf{0}$ and $k = 0$.
2. **repeat**
3. Calculate $\hat{\mathbf{q}} = \mathbf{X}'(\mathbf{y} - \hat{\mu}_k)$ and set $\hat{Q} = max_i \{|\hat{q}_i|\}$.
4. Let $\mathbb{A} = \left\{ i : |\hat{q}_i| = \hat{Q} \right\}$.
5. Set $\mathbf{X}_{\mathbb{A}} = (...\mathbf{x}_i...)_{i \in \mathbb{A}}$ for calculating $\bar{\mathbf{y}}_{k+1} = \left(\mathbf{X}'_{\mathbb{A}}\mathbf{X}_{\mathbb{A}}\right)^{-1} \mathbf{X}'_A\mathbf{y}$ and $\mathbf{a} = \mathbf{X}'_{\mathbb{A}}(\bar{\mathbf{y}}_{k+1} - \hat{\mu}_k)$.
6. Set
$$\hat{\mu}_{k+1} = \hat{\mu}_k + \hat{\gamma}\left(\hat{\mathbf{y}}_{k+1} - \hat{\mu}_k\right),$$
where, if $\mathbb{A}^c \neq \oslash$,
$$\hat{\gamma} = min^+_{i \in \mathbb{A}^c} \left\{ \frac{\hat{Q} - \hat{q}_i}{\hat{Q} - \hat{a}_i}, \frac{\hat{Q} + \hat{q}_i}{\hat{Q} + \hat{a}_i} \right\},$$
otherwise set $\hat{\gamma} = 1$.
7. $k \leftarrow k + 1$.
8. **Until** $\mathbb{A}^c = \oslash$.

It can be shown that, with one modification, this procedure gives the entire path of LASSO solutions, as the penalty is varied from 0 to infinity. The modification needed is: if a coefficient crosses zero, stop. Drop that predictor, recompute the best direction and continue. This gives the LASSO path.

1.3 Our Work and Contribution

In this section, we give a short motivation and the used methods and obtained results for each of the chapters.

1. **Sparse Gaussian Graphical Mixture Models**

   Motivation. Biologists are interested in the dependency structure among large network of genes. This is often done without taking into consideration the heterogeneity nature of the sample. By heterogeneity, we mean that networks may be different for different samples of observations. Stated differently, individuals in the population are rarely homogenous and may come from several distinct subpopulations each with their own underlying dependency structure. However, typically little information is known about an individual's subpopulation membership. The question now is how to model such heterogeneity and recover the underlying networks from which the clusters of samples originate from?

   Methods and Results. Statistical methods for analyzing such data are subject to active research currently (Agakov et al., 2012). In this chapter we propose Gaussian graphical mixture models (GGMM) to model such data. In this particular context, it is well known that parameter estimation is challenging due to large number of variables coupled with the degenerate nature of the likelihood function. We propose as a solution a penalized Gaussian graphical mixture model by imposing an L_1 penalty on the precision matrix. Our approach shrinks the covariance matrices thereby resulting in better identifiability and variable selection. We adopt an Expectation Maximization (EM) algorithm which involves the graphical least absolute shrinkage and selection operator (GLASSO) to estimate the networks. We show that under certain regularity conditions the Penalized Maximum Likelihood (PML) estimates are consistent. The corresponding R- package is included in appendix A.

   Examples. We demonstrate the performance of the PML estimator through simulations and we show the utility of our method for real data analysis. Two different schemes based on the choice of the regularization parameters are investigated at the simulation stage to demonstrate the consistency property of our PMLE. Our method has also been applied to 2 real data sets.

2. **State space modelling of dynamic genetic networks**

   Motivation. The genomic reality is highly complex and dynamic. The recent development of high-throughput technologies has enabled researchers to measure the abundance of thousands genes through time. The challenge is to unravel from such measurements, gene-protein or gene-gene or protein-protein interactions and key biological features of cellular systems. We devise a method for inferring transcriptional or gene regulatory networks from high-throughput data sources such as gene expression microarrays with potentially hidden states,

such as unmeasured transcription factors (TFs), which satisfies certain Markov properties.

Methods and Results. In an attempt to account for the effects of such hidden states, we build a mathematical model, able to capture the stochastic nature of the biological process as well as their dynamics behavior. We assume the observations are noisy measurements of gene expression in the form mRNAs whose dynamics can be described by some hidden process and build a dynamic state space representation from these hidden states. Our method is based on an EM algorithm with an incorporated Kalman smoothing algorithm in the E-step to calculate the hidden states. We obtain an explicit formulation of the parameters defining our state space model, and provide means for constructing reliable gene regulatory networks based on bootstrap statistical analysis. We adopt Akaike Information Criterion (AIC) for model selection. The state space model is an approach with proven effectiveness to reverse engineer transcriptional networks.

Examples. The proposed method is applied to time course microarray data obtained from a well-established T-cell experiment. Our results support interesting biological properties in the family of Jun genes. Regulatory genes include JUND proto-oncogene, the cell division cycle 2 (CDC2), the FYN-binding protein gene (FYB). We found an interaction between JUNB and SMN1 and discovered that JUND activates CDC2.

3. **SSM with L_1 regularization constraint**

Motivation. Microarray technologies and related methods coupled with appropriate mathematical or statistical models have made it possible to identify dynamic regulatory networks by measuring time course expression levels of many genes simultaneously. However one of the challenges relate to the high-dimensional nature of such data in addition to the fact that these gene expression data are known to exclude some hidden process.

Methods and Results. We build an input-dependent penalized linear state space model from these hidden states and propose an L_1 penalized inference approach. We demonstrate how an incorporated L_1 regularization constraint in an Expectation-Maximization (EM) algorithm can be used to reverse engineer transcriptional networks from gene expression profiling data. Penalized maximum likelihood estimates were obtained for the penalized state space model through a simple modification of the Least Angle Regression (LARS) algorithm. Parameters become identifiable as a result of the L_1 penalty. This allows useful

interpretations of the model.

Examples. We perform in silico experiment using Arabidopsis thaliana clock data to validate our method. The proposed method is also illustrated on time-course microarray data obtained from a well established T-cell experiment. At the optimum tuning parameters we found genes TRAF5, JUND, CDK4, CASP4, CD69, and C3X1 to have higher number of inward directed connections and FYB, CCNA2, AKT1 and CASP8 to be genes with higher number of outwards directed connections. Caspase 4 is also found to activate the expression of JUND which in turn represses the cell cycle regulator CDC2.

1.4 Organization of the Chapters

After this introduction which was aimed at outlining the core of the contributions of this thesis, outlining networks estimating models as well as the statistical inference methods which will be employed. The rest of the dissertation is organized as follows:

Chapter 2 describes sparse Gaussian graphical mixture models for networks reconstruction. The latter is supported by an R package `glassomix` developed for multiple networks discovery. (See appendix A).

A gene regulatory network (GRN) is a complex system, which is appropriately modeled in a dynamic way. Chapter 3 gives an in-depth discussion of the linear state space model, introduces some mathematical interpretations, and extend it in modelling and inferring gene regulation through estimation of state parameters and state dynamics. We explicitly estimate the parameters from the model via an expectation-maximization algorithm. Classical statistical inferences were performed through bootstrap statistical analysis and edge selection or deletion is done through hypothesis testing at some level α.

However, many current gene expression data sets include a large number of genes, but only few samples (large p, small n). This problem demands care in the estimation of model parameters in the SSM. Parameter identifiability is also another complication. In chapter 4, we carry on with the SSMs and give it a precise biological interpretation. We then introduce a regularization technique through L_1 penalized inference. This gives rise to penalized state space models (PSSM). The proposed method in the maximization step of the EM-algorithm is the L_1 penalty through a simple modification of the LARS algorithm by Efron et al. (2004), (Least Angle Regression). LARS is an efficient algorithm for computing the entire regularization path for the Lasso.

Chapter 2

Sparse Gaussian Graphical Mixture Model

2.1 Introduction

Biologists aim to describe the dependency structure among large number of genes. This is often done without taking into consideration the heterogeneity nature of the samples. By heterogeneity, we mean networks may be different for different subgroups of samples. Our population of individuals may come from several distinct subpopulations each with their own underlying dependency structure. However, typically little information is known about an individual's group membership. In this setting, parameters may change for different subgroups of observations. We want to model such heterogeneity and recover the underlying networks from such data with some sparsity constraint. The problem becomes more complex if the number of components that made up the population is unknown. Statistical methods for analyzing such data are subject to active research currently (Agakov et al., 2012). Gaussian graphical mixture models (GGMM) are ways to model such data.

A Gaussian graphical model (GGM) for a random vector $Y = (Y_1, ..., Y_p)$ is a pair $(\mathbb{G}, \mathbb{P})$ where $\mathbb{G}$ is an undirected graph and $\mathbb{P} = \{N(\mu, \Theta^{-1})\}$ is the model comprising all multivariate normal distributions whose inverse covariance matrix or precision matrix entries satisfies $(u, v) \in \mathbb{G} \iff \Theta_{uv} \neq 0$. The conditional independence relationship among nodes are captured in the precision matrix Θ. Consequently, the problem of selecting the graph is equivalent to estimating the off-diagonal zero-pattern of the concentration matrix. Further details on these models as well interpretation of the conditional independency on the graph can be found in (Lauritzen, 1996).

Mixture distributions are often used to model heterogeneous data or observations supposed to have come from one of K different networks or components. Under Gaussian mixtures, each component is suitably modelled by a family of Gaussian

probability density. This chapter deals with the problem of structural learning in reconstructing the underlying graphical networks (using a graphical Gaussian model) from a data supposed to have come from a mixture of Gaussian distributions.

We consider model-based clustering (McLachlan et al., 2002) and assume that the data come from a finite mixture model where each component represents a network. A large literature exists in normal mixture models; (Lo et al., 2001; Bozdogan, 1983). Our focus here is on a high dimensional data setting where we present an algorithm based on a regularized expectation maximization using Gaussian mixture model (GMM). We assume that our data $\mathbf{Y}_i = (Y_{i1}, ..., Y_{ip})^{'}$ is generated through $K \geq 1$ latent generative mixture components. We aim to group the data into a few K networks and identify which observations are from which Gaussian components.

A natural way for parameter estimation in GMMs is via a maximum likelihood estimation. However some performance degradation is encountered owing to the identifiability of the likelihood and the high dimensional setting. To overcome these problems, Banfield and Raftery (1993) proposed a parameter reduction technique by re-parameterizing the covariance matrix through eigenvalue decomposition. In doing so, some parameters are shared across clusters. As a result of a continuous increasing number of dimensions, this approach can not totally alleviate the ($n << p$) phenomena. Recently proposals to overcome the high dimensionality problem involve estimating sparse precision matrix. Among these proposals is the penalized log likelihood technique of Friedman et al. (2008a), an L_1 regularization approach which encourages many of the entries of the precision matrix to be 0. Our method is based on this idea. The L_1 penalty promotes sparsity. We provide sufficient conditions for consistency of the penalized MLE.

Closely related to our work is that of Pan and Shen (2007) where variable selection is considered in model-based clustering. They considered GMM and penalize only the mean vectors and seeking to estimate sparse mean vectors. They assumed a common diagonal covariance matrix for all clusters. This work was later extended to (Zhou et al., 2009) where a new approach to penalized model-based clustering was considered but this time with unconstrained covariance matrices. However not much has been said about the consistency of the resulting estimators. Another recent work in this field is the work by Agakov et al. (2012) that learn structures of sparse high dimension latent variables with application to mixtures.

In this chapter we propose a penalized likelihood approach in the context of Gaussian graphical mixture model, which constraints the networks to be sparse. The parameters in the networks are estimated by incorporating an existing Graphical LASSO (GLASSO) method for covariance estimation into an EM algorithm. In effect, we view each network as an instance of a particular GGM. Therefore we aim at

recovering the underlining various networks from which the data originate from. Additionally, we assess how well the resultant graphs obtained through GLASSO relate to the true graphs and we provide consistency results of the estimates. Throughout this chapter we assume K, the number of components of mixture models is known.

The reminder of this chapter is organized as follows. We introduce the model, set up the Penalized Maximum Likelihood Estimate (PMLE) approach and summarize the main result in connection with the consistency of the estimates obtained from the mixture model in section 2.2. We then proceed with the inference procedure through a penalized version of the EM algorithm in section 2.3. In section 2.4 we present some simulations and an example of applications to illustrate our results. We conclude with a brief discussion and future works in section 2.5.

2.2 Penalized maximum likelihood estimation

In this section we introduce the Gaussian Graphical Mixture Model, then we derive the penalized likelihood upon which statistical inference via the EM algorithm is based and prove consistency of the Penalized Maximum Likelihood Estimates (PMLE).

2.2.1 The Mixture model

Mixture models are very popular for the analysis of complex data. A mixture model represents the given data as a mixture of K networks or components, each of which has different characteristics. We introduce our model in Figure (2.1), where we assume a genetic population. We suppose sample of expression level of these genes comes from two different networks after observing their metabolism structure. We then fit two Gaussian distribution $N(\mu_1, \Theta_1)$ and $N(\mu_2, \Theta_2)$ for these clusters. Figure (2.1) represents the above via a mixture model. The question now is how can we infer the underlying networks from which the data come from?

Suppose we are given a training data set $\{\mathbf{Y}_1, ..., \mathbf{Y}_n\}$, assumed to be a random sample from K mixture components. Our model consists of assuming that the variable Z_i, describing which network an individual originates, is a multinomial random variable with parameters, π_k, denoting the mixture proportions or the mixing coefficients with $(0 < \pi_k < 1)$, $\sum_{k=1}^{K} \pi_k = 1$, and K is known. In essence

$$P(Z_i = k) = \pi_k.$$

We wish to model the data by specifying a joint distribution

$$P(\mathbf{Y}_i, Z_i) = P(\mathbf{Y}_i | Z_i) p(Z_i).$$

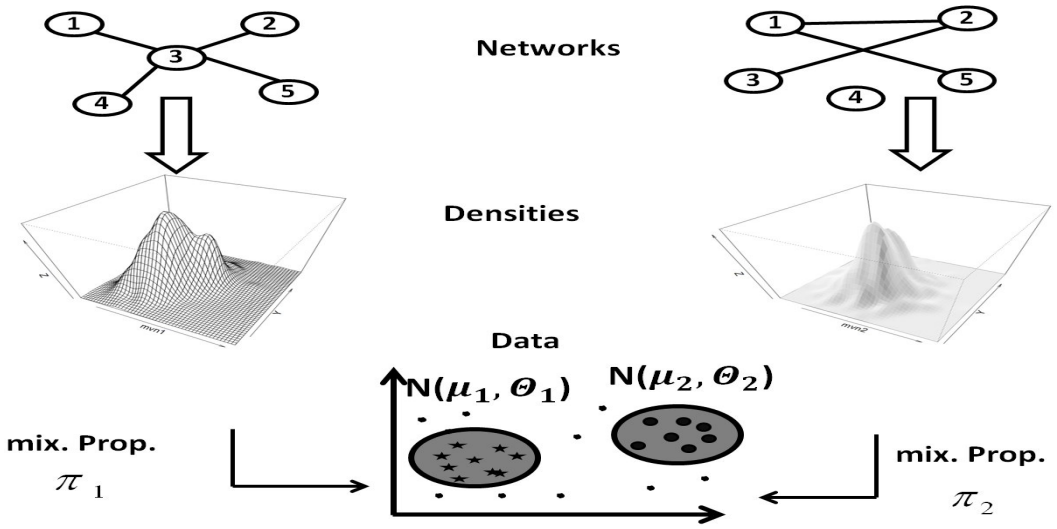

Figure 2.1. Mixture models: we assume the data is composed of 2 separate mixtures of Gaussian (MOG), each with a corresponding graphical model or network.

We model each subpopulation separately by assuming a GGM where $(\mathbf{Y}_i|Z_i = k) \sim N(\mu_k, \Sigma_k)$. Our model posits that each $\mathbf{Y}_i$ was generated by randomly choosing Z_i from $\{1, ...K\}$, or $\mathbf{Y}_i$ was drawn from one of the k Gaussian depending on Z_i.

In this work we assume that $\forall\ k$, $\mu_k = 0$. In practice, this means that the data is assumed to be normalized by subtracting the mean. Since $\mathbf{Y}_i$ is dependent on Z_i, we say that Z_i represents the class that produced $\mathbf{Y}_i$ and we know $\mathbf{Y}_i$ fully if we know which class Z_i falls. Also note that the Z_i's are latent random variables, meaning that they are hidden or unobserved. The density of each $\mathbf{Y}_i$ can be written as

$$\begin{aligned} f_\gamma(\mathbf{y}_i) &= \sum_{k=1}^{K} \pi_k \varphi_k(\mathbf{y}_i|Z_i = k) \\ f_\gamma(\mathbf{y}_i) &= \sum_{k=1}^{K} \pi_k \varphi_k(\mathbf{y}_i|\Theta_k) \end{aligned} \tag{2.1}$$

where $\varphi(\mathbf{y}_i|\Theta_k)$ denotes the density of Gaussian distribution with mean 0 and inverse covariance covariance matrix Θ_k; f_γ represents the "incomplete" mixture data density

of the sample i.e $\mathbf{y} \sim f_\gamma$. We introduce the parameter set of mixture namely

$$\Omega = \left\{ \{\Theta_k\}_{k=1}^K | \Theta_k \succ 0, \quad k = 1, ..., K \right\},$$

$\Theta \succ 0$ indicates that Θ is positive-definite matrix, and

$$J = \left\{ \{\pi_k\}_{k=1}^K | \pi_k > 0, \quad k = 1, ..., K \right\}$$

and

$$\Gamma = \Omega \times J \tag{2.2}$$

denotes the parameter space with the true parameter defined as $\gamma_0 = (\Theta_0, \pi_0) \in \Gamma$.

In order to characterize the mixture model and estimate its parameters thereby recovering the underlying graphical structure from the data (seen as mixture of multivariate densities), several approaches may be considered. These approaches include graphical methods, methods of moments, minimum-distance methods, maximum likelihood (Ruan et al., 2011; Zhou et al., 2009) and Bayesian methods (Bernardo, 2003; Biernacki et al., 2000). In our case we adopt the penalized maximum likelihood method in a graphical model set up.

2.2.2 The penalized model-based likelihood

We can now write the likelihood of the incomplete data density as

$$L_{\mathbf{y}}(\gamma) = \prod_{i=1}^{n} \left[\sum_{k=1}^{K} \pi_k \varphi_k(\mathbf{y}_i | \Theta_k^{-1}) \right],$$

whose log-likelihood function is given by

$$l_{\mathbf{y}}(\gamma) = \sum_{i=1}^{n} \log f_\gamma(\mathbf{y}_i) \tag{2.3}$$

The goal is to maximize the log-likelihood in (2.3) with respect to γ. Unfortunately, a unique global maximum likelihood estimate does not exist because of the permutation symmetries of the mixture subpopulation; (Day, 1969; Surajit and Lindsay, 2005). Also the likelihood function of normal mixture models is not a bounded function on γ as was put forward by Kiefer and Wolfowitz (1956). On the question of consistency of the MLE, Chanda (1954), Cramer (1946) focus on local ML estimation and mathematically investigate the existence of a consistent sequence of local

maximizers. These results are mainly based on Wald's technique (Wald, 1949). Redner (1981) later extended these results to establish the consistency of the MLE for mixture distributions with restrained or compact parameter spaces. It was proved that the MLE exists and it is globally consistent in a compact subset $\hat{\Gamma}$ of Γ that contains γ_0; i.e

$$\text{given} \quad \hat{\gamma}_n | l_{\mathbf{y}}(\hat{\gamma}_n) = \max_{\gamma \in \Gamma} \quad l_{\mathbf{y}}(\gamma), \quad \hat{\gamma}_n \to \gamma_0 \text{ in probability, for n} \to \infty$$

In addition to the degenerate nature of the likelihood (Kiefer and Wolfowitz, 1956) on the set Γ, the "high dimensional, low sample size setting"- where the number of observations n is smaller that the number of nodes or features p- is another complication. Estimating the parameters in the GGMM by maximizing criterion (2.3) is a complex one. The penalized likelihood-based method (Friedman et al., 2008a; Yuan and Lin, 2007) is a promising approach to counter the degeneracy of $l_{\mathbf{y}}(\gamma)$ while keeping the parameter space Γ unaltered. However, to make the PMLE work, one has to solve the problem of what kind of penalty functions are eligible. We opt for a penalty function that prevents the likelihood from degenerating under the multivariate mixture model. We assume that the penalty function $P : \Gamma \to \mathbb{R}_0^+$ given by

$$P(\Theta) = \exp(-\lambda ||\Theta||_1),$$

satisfies:

$$\lim_{|\Theta_k| \to \infty} P(\Theta_k) |\Theta_k|^n = 0 \quad \forall k \in \{1, 2, ..., K\} \quad \forall n \tag{2.4}$$

where $\lambda > 0$ is a user-defined tuning parameter that regulates the sparsity level, $|\Theta|$ denotes determinant of Θ, and $||.||_1$ is the L_1 norm or the sum of absolute values of the entries of a matrix or a vector i.e $||\mathbf{X}||_1 = \sum_{i=1}^{n} |X_i|$.

This results in placing an L_1 penalty on the entries of the concentration matrices so that the resulting estimates are sparse and zeroes in these matrices correspond to conditional independency between the nodes similar to (Nicolai et al., 2006). Numerous advantages result from this approach. First of all, the corresponding penalized likelihood is bounded and the penalized likelihood function does not degenerate in any point of the closure of parameter space Γ and therefore the existence of the penalized maximum likelihood estimator is guaranteed. Next, in the context of GGM, penalizing the precision matrix results in better estimates and sparse models are more interpretable and often preferred in application.

We define the L_1 penalized log-likelihood as:

$$l_{\mathbf{y}}^{p}(\gamma) = l_{\mathbf{y}}(\gamma) - \lambda_n \sum_{k=1}^{K} ||\Theta_k||_1 \tag{2.5}$$

where $\lambda_n \propto \frac{\lambda}{\sqrt{n}}$, $||\Theta||_1 = \sum_{i,j} |\theta_{ij}|$, K is the number of mixing components assumed fixed. The hyperparameters K and λ determine the complexity of the model. The corresponding PMLE are defined as

$$\hat{\gamma}_{\lambda_n} = \arg\max_{\gamma} \quad l_y^p(\gamma) \tag{2.6}$$

Our method penalizes all the entries of the precision matrix including the diagonal elements. We do this in order to avoid the likelihood to degenerate. To see this, consider a special case of a model consisting of two univariate normal mixtures $\pi_1\varphi(\mathbf{y}|\sigma_1)+\pi_2\varphi(\mathbf{y}|\sigma_2)$. By letting $\sigma_1 \to 0$ with other parameters remaining constant, the log-likelihood tends to infinity for values of $y = 0$, i.e the log-likelihood degenerates due to mixture formulation whereby a single observation mixture component with a decreasing variance on top of the observation explodes the likelihood. For that matter an L_1 penalty which does not penalize the diagonal elements tend to result in a degenerate ML estimator especially when $n \to \infty$.

2.2.3 Consistency

At this stage we want to characterize the solution obtained in Equation (2.6). The general theorem concerning the consistency of the MLE (Redner, 1980; Wald, 1949) can be extended to cover our type of penalized MLE. This is because if a likelihood function which yields a strong consistent estimate over a compact set is given, then our L_1 penalty would not alter the consistency properties. Consistency of the PMLE is given in theorem 2.2.3. The latter uses results in (Wald, 1949) under the classical MLE over a compact set.

Before we present our result relating to the consistency of our PMLE, we summarize the corresponding MLE version in the following lemmas. First the following assumptions will be needed.

A1: There is a neighborhood ρ of γ_0 such that for all $\gamma \in \rho$; for almost all $\mathbf{y} \in R^n$; and for for l, j and $s = 1, ..., \upsilon$; $\frac{\partial f}{\partial \gamma_l}$, $\frac{\partial^2 f}{\partial \gamma_l \partial \gamma_j}$, $\frac{\partial^3 f}{\partial \gamma_l \partial \gamma_j \partial \gamma_s}$ exist and satisfy

$$\left|\frac{\partial f}{\partial \gamma_l}\right| < g_l(\mathbf{y}); \left|\frac{\partial^2 f}{\partial \gamma_l \partial \gamma_j}\right| < g_{lj}(\mathbf{y}); \left|\frac{\partial^3 f}{\partial \gamma_l \partial \gamma_j \partial \gamma_s}\right| < g_{ljs}(\mathbf{y}),$$

where g_l, and g_{lj} are integrable and $g_{ljs}(\mathbf{y})$ satisfies

$$\int_{R^n} g_{ljs}(\mathbf{y}) f_{\gamma_0}(\mathbf{y}) d\mathbf{y} < \infty.$$

A2: The matrix $\delta(\gamma) = \left(\int_{R^n} \frac{\partial \ln f}{\partial \gamma_l} \frac{\partial \ln f}{\partial \gamma_j} f d\mathbf{y} \right)$ is positive definite at γ_0.

Lemma 2.2.1. *If conditions A1 and A2 are satisfied, then, given any sufficiently small neighborhood ρ_0 of γ_0 with probability equals 1 as the sample size n approaches infinity, there is a unique solution to the likelihood equations in ρ_0 and this solution is an MLE.*

The proof of lemma 2.2.1 is provided in appendix B. Lemma 2.2.1 indicates that, by restricting attention to a fixed neighborhood of γ_0, we have a unique and consistent solution to the likelihood equations.

The next lemma considers a situation where the likelihood is an unbounded function. For that one must assume a compact (closed and bounded) parameter space. It will be assumed that there is a σ- finite measure μ such that for each $\gamma \in \Gamma$ the probability measure μ_γ is absolutely continuous w.r.t. μ. We let $f_\gamma(\mathbf{y})$ denote any representative of the density of μ_γ w.r.t μ. The following assumptions are made in addition:

C1: The parameter space Γ is a closed and bounded subset of R^l for some positive number l. In particular, $T = \{(\Theta_1, ..., \Theta_K)\} \mid \quad s.t. \quad ||\Theta_k||_1 \leq M^*$ and $||\Theta_k||_2 \geq \epsilon^*, k = 1, ..., K$, for some positive number M^* and ϵ^*.

C2: Let $B_r(\gamma)$ be the closed ball of radius r about γ. Then for any positive real number r, let:

$$f_\gamma(\mathbf{y}, r) = \sup_{\eta \in B_r(\gamma)} f_\gamma(\mathbf{y}, \eta); \quad f^*_\gamma(\mathbf{y}, r) = max\left[1, f_\gamma(\mathbf{y}, r)\right].$$

Then for each γ and for sufficiently small r

$$\int \ln f^*_\gamma(\mathbf{y}, r) d\mu_{\gamma_0} < \infty.$$

C3:

$$\int \left| \ln f_{\gamma_0}(\mathbf{y}) \right| d\mu_{\gamma_0} < \infty.$$

C4: if $\gamma_l \to \gamma$, then $f_{\gamma_l}(\mathbf{y}) \to f_\gamma(\mathbf{y})$.

Lemma 2.2.2. *Given assumptions (C1-C4), and let* $C = \{\gamma \in \Gamma | f_\gamma(\mathbf{y}) = f_{\gamma_0}(\mathbf{y}) \quad \textit{almost everywhere}\}$. *If* S *is any open neighborhood containing* C, *then with probability equals* 1, *the MLE is eventually in* S.

The 2 lemmas show that the MLE converges to the set C. Since C is the set of all parameters for which the density is the true density, it may be said that the MLE converges strongly to the true set of parameters.

We then define 2 further conditions upon which our theorem 2.2.3 holds.

C5: Let $\bar{\Gamma}$ denotes the quotient topological space obtained from Γ and suppose that $\bar{\Gamma}$ is any compact subset containing γ_0.

C6:
$$\int |\ln f_l(\mathbf{y}, \gamma_l)| \, d\mu_{\gamma_j} < \infty \quad for \quad \gamma_l \in \Gamma_l \quad \text{and} \quad \gamma_j \in \Gamma_j.$$

Theorem 2.2.3. *Suppose that the mixing distributions satisfy conditions (C1-C6). Define* $|\gamma_0| = ||\pi_0||_2 + ||\Theta_0||_F$. *Suppose that* π_k *is bounded away from zero, and the penalty is set as* $\lambda_n \propto (1/\sqrt{n})$. *It follows that for a fixed* p, *the penalized likelihood solution* $\hat{\gamma}_{\lambda_n}$ *is consistent in the quotient topological space* $\bar{\Gamma}$, *i.e* $\forall \epsilon > 0$

$$\lim_{n\to\infty} P\left(|\hat{\gamma}_{\lambda_n} - \gamma_0| > \epsilon\right) = 0.$$

Proof. Let the PMLE $\hat{\gamma}_{\lambda_n}$ and MLE $\hat{\gamma}_n$ be defined by

$$\hat{\gamma}_{\lambda_n} = \arg\max_{\gamma} \quad l_n^p(\gamma),$$

and

$$\hat{\gamma}_n = \arg\max \quad l(\gamma),$$

where

$$l_n^p(\gamma) = l(\gamma) - \lambda_n \sum_{k=1}^{K} ||\Theta_k||_1, \quad \forall \quad k \in \{1, ..., K\}.$$

Then

$\forall \epsilon > 0$ we have

$$\begin{aligned} P(|\hat{\gamma}_{\lambda_n} - \gamma_0| > \epsilon) &= P(|\hat{\gamma}_{\lambda_n} - \hat{\gamma}_n + \hat{\gamma}_n - \gamma_0| > \epsilon) \\ &\leq P(|\hat{\gamma}_{\lambda_n} - \hat{\gamma}_n| > \epsilon/2) + P(|\hat{\gamma}_n - \gamma_0| > \epsilon/2) \end{aligned} \tag{2.7}$$

Considering the second inequality on the right hand side of Equation (2.7), we have, from the consistency of the MLE, that

$$\lim_{n\to\infty} P\left(|\hat{\gamma}_n - \gamma_0| > \epsilon/2\right) = 0.$$

Therefore it is sufficient to prove that

$$\lim_{n\to\infty} P(|\hat{\gamma}_{\lambda_n} - \hat{\gamma}_n| > \epsilon/2) = 0.$$

Suppose $l_n^p(\gamma)$ is bounded from below by a function $l_{n,L}^p(\gamma)$ given as

$$l_{n,L}^p(\gamma) = l(\gamma) - \lambda_n^* ||\Theta||_2, \quad \text{where} \quad \lambda_n^* = \lambda_n \frac{M^*}{\epsilon^*},$$

M^* and ϵ^* are given in C1. Then

1. There exists a neighborhood γ_0 of Γ such that $l_{n,L}^p(\gamma)$ is continuously differentiable with respect to parameters in γ
2. $l_{n,L}^p(\gamma)$ converges (pointwise) to $l(\gamma)$ as $n \to \infty$.

We define

$$\hat{\gamma}_{\lambda_{n,L}} = \arg\max_{\gamma} \quad l_{n,L}^p(\gamma).$$

Then the following holds:

$$\forall \delta > 0 \quad \exists \quad n_1 \in N \quad s.t. \quad \forall n > n_1, \quad |\hat{\gamma}_{\lambda_{n,L}} - \hat{\gamma}_n| < \delta.$$

For the MLE, we have $\frac{\partial^2 l}{\partial \gamma^2}(\hat{\gamma}_n) = O_p(nM)$ is negative definite, where M is a constant matrix that depends on γ_0. So for a fixed n_0, and $\forall n > n_0$ we have

$$|l(\gamma) - l_{n_0,L}^p(\gamma)| > |l(\gamma) - l_{n,L}^p(\gamma)| \quad \forall \quad \gamma.$$

Therefore

$$\exists \quad n_2 \in N \quad s.t. \quad \forall n > n_2, \quad and \quad \forall \gamma \in B_{\frac{\epsilon}{4}}(\hat{\gamma}_n),$$

$$l(\gamma) - l^p_{n,L}(\hat{\gamma}_{\lambda_{n,L}}) \geq 0.$$

In particular, as $n \to \infty$ and for $\lambda_n \propto (1/\sqrt{n})$. we have

$$l^p_n(\hat{\gamma}_{\lambda_n}) - l(\gamma) = l^p_{n,L}(\hat{\gamma}_{\lambda_{n,L}}) - l(\gamma) \to 0,$$

then $\forall \epsilon > 0$, we have

$$P(|\hat{\gamma}_{\lambda_n} - \hat{\gamma}_n| > \frac{\epsilon}{2}) \;\; \leq \;\; P(|\hat{\gamma}_{\lambda_n} - \hat{\gamma}_{\lambda_{n,L}}| > \frac{\epsilon}{4}) + P(|\hat{\gamma}_{\lambda_{n,L}} - \hat{\gamma}_n| > \frac{\epsilon}{4}) \tag{2.8}$$

But

$$\lim_{n\to\infty} P\left(|\hat{\gamma}_{\lambda_n} - \hat{\gamma}_{\lambda_{n,L}}| > \frac{\epsilon}{4}\right) = 0.$$

□

2.3 Penalized EM algorithm

In order to maximize the penalized likelihood function (2.5) we consider a penalized version of the EM algorithm of Dempster et al. (1977). To do that we first augment our data $\mathbf{Y}_i$ with $\mathbf{Z}_i$ so that the complete data associated with our model now becomes $\mathbf{C}_i = (\mathbf{Y}_i, \mathbf{Z}_i)$ and an EM algorithm iteratively maximizes, instead of the penalized observed log-likelihood $l^p_{\mathbf{y}}$ in (2.5), the conditional expectation of the penalized log-likelihood of the augmented data and $\Omega^{(t)}$ is the current value at iteration t.

Suppose $\mathbf{c}_i \sim h_{\mathbf{c}_i}(\gamma)$, i.e $h_{\mathbf{c}_i}(\gamma)$ is the density of the augmented data $\mathbf{c}_i$. Now the penalized log-likelihood of the augmented data can be written as

$$\begin{aligned}
l^p_{\mathbf{c}}(\gamma) &= \ln\left[h_{\mathbf{c}_i}(\gamma)\right] - \lambda \sum_{k=1}^{K} ||\Theta_k||_{l_1} \\
l^p_{\mathbf{c}}(\gamma) &= \sum_{i=1}^{n} \left(\ln \pi_k + \ln \varphi_k(\mathbf{y}_i|\Theta_k^{-1})\right) - \lambda \sum_{k=1}^{K} ||\Theta_k||_{l_1} \\
&= \sum_{i=1}^{n}\sum_{k=1}^{K} 1_{\{Z_i=k\}} \left[\ln \pi_k + \ln \varphi_k(\mathbf{y}_i|\Theta_k^{-1})\right] - \lambda \sum_{k=1}^{K} ||\Theta_k||_{l_1}
\end{aligned} \tag{2.9}$$

The indicator function $1_{\{Z_i=k\}}$ simply says that if you knew which component the observation i came from, we would simply use its corresponding Θ_k for the likelihood.

For illustration purpose, and suppose we have 3 observations and we are certain that the first two were generated by the Gaussian density $N(0, \Theta_2)$, and the last came from $N(0, \Theta_1)$. Then we write the full log-likelihood as follows:

$$l_{\mathbf{c}}(\Theta) = l_{\mathbf{y}_1}(\Theta_2) + l_{\mathbf{y}_2}(\Theta_2) + l_{\mathbf{y}_3}(\Theta_1) \tag{2.10}$$

2.3.1 The E-step

From Equation (2.9) we compute the quantity $Q(\gamma|\gamma^{(t)})$ as follows

$$\begin{aligned} Q(\gamma|\gamma^{(t)}) &= E_{\mathbf{Z}_i}\left[l_{\mathbf{c}}(\gamma) - \lambda||\Theta||_1|\mathbf{y}; \gamma^{(t)}\right] \\ &= \sum_{i=1}^{n}\sum_{k=1}^{K}\left[\ln \varphi_k(\mathbf{y}_i|\Theta_k^{-1}) + \ln \pi_k\right] E_{\mathbf{Z}_i}\left[1_{\{Z_i=k\}}|\mathbf{y}_i; \gamma^{(t)}\right] - \lambda||\Theta_k||_1 \\ &= \sum_{i=1}^{n}\sum_{k=1}^{K}\left[\ln \varphi_k(\mathbf{y}_i|\Theta_k^{-1}) + \ln \pi_k\right] P\left(Z_i = k|\mathbf{y}_i; \gamma^{(t)}\right) - \lambda||\Theta_k||_1 \\ &= \sum_{i=1}^{n}\sum_{k=1}^{K}\left[\ln \varphi_k(\mathbf{y}_i|\Theta_k^{-1}) + \ln \pi_k\right] \omega_{ik}^{(t)} - \lambda||\Theta_k||_1 \end{aligned} \tag{2.11}$$

The E-step actually consists of calculating ω_{ik}, the probabilities (conditional on the data and $\gamma^{(t)}$) that $\mathbf{Y}_i$'s originate from component k. It can also be seen as the responsibility that component k takes for explaining the observation $\mathbf{Y}_i$ and it tells us for which group an individual actually belongs. Using Bayes theorem, we have:

$$\begin{aligned} \omega_{ik}^{(t)} &= P\left(Z_i = k|\mathbf{y}_i, \gamma^{(t)}\right) \\ &= \frac{P(\mathbf{y}_i|Z_i = k; \gamma^{(t)})P(Z_i = k, \gamma^{(t)})}{\sum_{l=1}^{K} P(\mathbf{y}_i|Z_i = l; \gamma^{(t)})P(Z_i = l, \gamma^{(t)})} \\ &= \frac{\varphi_k^{(t)}(\mathbf{y}|\Theta_k^{-1})\pi_k^{(t)}}{\sum_{l=1}^{K} \varphi_l^{(t)}(\mathbf{y}_i|\Theta_k^{-1})\pi_l^{(t)}} \end{aligned} \tag{2.12}$$

2.3.2 The M-step

The M-step for our mixture model can be split in to two parts, the maximization related to π_k and the maximization related to Θ_k.

1. M-step for π_k:

   For the maximization over π_k we make use of the constraint that $\sum_{k=1}^{K} \pi_k = 1$ i.e $\pi_K = 1 - \sum_{k=1}^{K-1} \pi_k$ and $\pi_k > 0$. It turns out that there is an explicit form for π_k. Let $k_0 \in \{1, ..., K-1\}$. Then

$$\frac{\partial Q}{\partial \pi_{k_0}} = \sum_{i=1}^{n} \left[\frac{\omega_{ik_0}^{(t)}}{\pi_{k_0}} - \frac{\omega_{iK}^{(t)}}{1 - \sum_{k=1}^{K-1} \pi_k} \right] \tag{2.13}$$

   Setting $\frac{\partial Q}{\partial \pi_{k_0}} = 0$, yields the following:

$$\omega_{.k_0}^{(t)} \sum_{k=1}^{K-1} \pi_k + \pi_{k_0} \omega_{.K}^{(t)} = \omega_{.k_0}^{(t)} \tag{2.14}$$

   It can be shown that a unique solution to Equation (2.14) is

$$\begin{aligned} \pi_{k_0}^{(t+1)} &= \omega_{.k_0}^{(t)}/n \\ &= \sum_{i=1}^{n} \omega_{ik_0}^{(t)}/n \end{aligned} \tag{2.15}$$

2. M-step for Θ_k:

   Next, to maximize (2.11) over Θ_k, we only need the term that depends on Θ_k. The first thing we do here is to try to formulate the maximization problem for a mixture component to be similar to that for Gaussian graphical modeling with the aim of applying graphical LASSO method. The latter applies LASSO penalty to the inverse covariance matrix Θ with the aim of estimating sparse graphs.

The Graphical Lasso penalty

The graphical LASSO of Friedman et al. (2008a) is a regularization framework for estimating the covariance matrix. It is a sparse precision matrix estimation and is employed to discover which variables are affecting each other. The LASSO reduces the complexity of the model by forcing less influential variables to have no influence on the model. We now introduce the Graphical LASSO idea.

In what follow we let $\mathbf{W} = \mathbf{\Theta}^{-1}$ denote the covariance matrix. The graphical LASSO problem minimizes an L_1- regularized negative log-likelihood:

$$\underset{\mathbf{\Theta} \succ 0}{minimize} f(\mathbf{\Theta}) := -\ln\det(\mathbf{\Theta}) + tr(\mathbf{S\Theta}) + \lambda \sum_l \sum_s |\theta_{ls}| \tag{2.16}$$

where $\mathbf{S} = \frac{1}{n}\sum_{i=1}^{n}(\mathbf{y}_i - \mu)(\mathbf{y}_i - \mu)'$. The subgardient is set to zero and the solution of (2.16) satisfies

$$-\mathbf{\Theta}^{-1} + \mathbf{S} + \lambda \mathbf{G} = 0 \tag{2.17}$$

where g_{ls} is the sub-gradient of the function $|\theta_{ls}|$, or a matrix of component wise signs of $\mathbf{\Theta}$:

$$g_{ls} = sign(\theta_{ls}) \quad \text{if} \quad \theta_{ls} \neq 0$$

$$g_{ls} \in [-1, 1] \quad \text{if} \quad \theta_{ls} = 0$$

From the global stationary conditions of (2.17) we require that $\theta_{ss} > 0$, and this lead to

$$W_{ss} = S_{ss} + \lambda \quad s = 1, ..., p.$$

GLASSO uses a block-coordinate method for solving (2.17). Consider a partitioning of $\mathbf{\Theta}$ and $\mathbf{G}$:

$$\mathbf{\Theta} = \begin{pmatrix} \mathbf{\Theta}_{11} & \theta_{12} \\ \theta_{21} & \theta_{22} \end{pmatrix}, \quad \mathbf{G} = \begin{pmatrix} \mathbf{G}_{11} & \mathbf{g}_{12} \\ \mathbf{g}_{21} & g_{22} \end{pmatrix}$$

where $\mathbf{\Theta}_{11}$ is $(p-1) \times (p-1)$, θ_{12} is $(p-1) \times 1$ and θ_{22} is scalar. $\mathbf{W}$ and $\mathbf{S}$ are also partitioned in the same manner. Using properties of inverses of block-partitioned matrices, we have

$$\begin{pmatrix} \mathbf{W}_{11} & \mathbf{w}_{12} \\ \mathbf{w}_{21} & w_{22} \end{pmatrix} = \begin{pmatrix} \tilde{\mathbf{W}}_{11} & -\mathbf{W}_{11}\frac{\theta_{12}}{\theta_{22}} \\ . & \theta_{22}^{-1} - \frac{\theta_{21}\mathbf{W}_{11}\theta_{12}}{\theta_{22}^2} \end{pmatrix} \tag{2.18}$$

$$= \begin{pmatrix} \tilde{\mathbf{W}}_{11} & -\frac{\mathbf{\Theta}_{11}^{-1}\theta_{12}}{\theta_{22} - \theta_{21}\mathbf{\Theta}_{11}^{-1}\theta_{12}} \\ . & \tilde{\mathbf{W}}_{22} \end{pmatrix} \tag{2.19}$$

where $\tilde{\mathbf{W}}_{11} = \mathbf{\Theta}_{11}^{-1} + \frac{\mathbf{\Theta}_{11}^{-1}\theta_{12}\theta_{21}\mathbf{\Theta}_{11}^{-1}}{(\theta_{22}-\theta_{21}\mathbf{\Theta}_{11}^{-1}\theta_{12})}$ and $\tilde{\mathbf{W}}_{22} = (\theta_{22} - \theta_{21}\mathbf{\Theta}_{11}^{-1}\theta_{12})^{-1}$. We also note from (2.18) that

$$\theta_{12} = -\mathbf{W}_{11}\mathbf{w}_{12}\theta_{22}^{-1} \tag{2.20}$$

Equation (2.17) implies:

$$-\mathbf{w}_{12} + \mathbf{s}_{12} + \lambda\mathbf{g}_{12} = 0 \tag{2.21}$$

and plugging (2.20) into (2.21), we have:

$$\mathbf{W}_{11}\theta_{12}\theta_{22}^{-1} + \mathbf{s}_{12} + \lambda\mathbf{g}_{12} = 0 \tag{2.22}$$

GLASSO operates on the above gradient equation. The algorithm solves (2.22) for $\mathbf{b} = \theta_{12}\theta_{22}^{-1}$, that is

$$\mathbf{W}_{11}\mathbf{b} + \mathbf{s}_{12} + \lambda\mathbf{g}_{12} = 0 \tag{2.23}$$

Equation (2.23) is the stationary equation for the following L_1 regularized quadratic program:

$$\underset{\mathbf{b}\in\mathbb{R}^{p-1}}{minimize}\left\{\frac{1}{2}\mathbf{b}^{'}\mathbf{W}_{11}\mathbf{b} + \mathbf{b}^{'}\mathbf{s}_{12} + \lambda||\mathbf{b}||_1\right\} \tag{2.24}$$

This is a traditional LASSO regression problem for $\mathbf{b}$.

Now from Equation (2.11), for a specific cluster k_0, the term that depends on the cluster specific covariance matrix Θ_{k_0} is given by

$$\begin{aligned}
Q\left(\Theta_{k_0}\right) &= \sum_{i=1}^{n}\omega_{ik_0}^{(t)}\ln\varphi_{k_0}(\mathbf{y}_i|\Theta_{k_0}^{-1}) - \lambda||\Theta_{k_0}||_1 \\
&= \sum_{i=1}^{n}\omega_{ik_0}^{(t)}\left[\frac{1}{2}\ln|\Theta_{k_0}| - \frac{1}{2}\mathbf{y}_i^{'}\Theta_{k_0}\mathbf{y}_i\right] - \lambda||\Theta_{k_0}||_1 \\
&= \sum_{i=1}^{n}\frac{\omega_{ik_0}^{(t)}}{2}\ln|\Theta_{k_0}| - \frac{1}{2}tr\left(\sum_{i=1}^{n}\omega_{ik_0}^{(t)}(\mathbf{y}_i\mathbf{y}_i^{'})\Theta_{k_0}\right) - \lambda||\Theta_{k_0}||_1 \\
&= \frac{\omega_{.k_0}^{(t)}}{2}\left[\ln|\Theta_{k_0}| - tr\left(\tilde{S}_{k_0}\Theta_{k_0}\right) - \frac{2\lambda}{\omega_{.k_0}^{(t)}}||\Theta_{k_0}||_1\right] \\
&= \frac{\omega_{.k_0}^{(t)}}{2}\left[\ln|\Theta_{k_0}| - tr\left(\tilde{S}_{k_0}\Theta_{k_0}\right) - \lambda_n||\Theta_{k_0}||_1\right]
\end{aligned} \tag{2.25}$$

where

$$\omega_{.k_0}^{(t)} = \sum_{i=1}^{n} \omega_{ik_0}^{(t)}$$

$$\tilde{S}_{k_0} = \frac{\sum_{i=1}^{n} \omega_{ik_0}^{(t)} (\mathbf{y}_i \mathbf{y}_i^{'})}{\omega_{.k_0}^{(t)}} \tag{2.26}$$

is the weighted empirical covariance matrix, and

$$\hat{\Theta}_{k_0} = \arg\max_{\Theta} \left\{ \ln |\Theta_{k_0}| - tr(\tilde{S}_{k_0} \Theta_{k_0}) - \lambda_n ||\Theta_{k_0}||_1 \right\} \tag{2.27}$$

subject to the constraint that Θ_{k_0} is positive definite with $\lambda_n = \frac{2\lambda}{\omega_{.k_0}^{(t)}}$.

Therefore the maximization of Θ_k consists of running the graphical LASSO procedure (Friedman et al., 2008a) for each cluster where each observation $\mathbf{Y}_i$ for Θ_k gets a weight and the sampling covariance matrix S_k is transformed to a weighted sampling covariance. This is a major innovation in our work where we formulate the Gaussian mixture modelling problem in a Gaussian graphical modelling framework. We summarize the algorithm below:

Initialize $\pi_1, ..., \pi_{Kmax}$, $\Theta_1, ..., \Theta_{Kmax}$
repeat
for $\lambda \in (\lambda_1, ..., \lambda_K)$
Compute:

1. E-step: $\omega_{ik} = \frac{\varphi_k(\mathbf{y}|\Theta_k^{-1})\pi_k}{\sum_{l=1}^{K} \varphi_l(\mathbf{y}_i|\Theta_k^{-1})\pi_l}$
2. M-step:
   - $\hat{\pi}_k = \sum_{i=1}^{n} \omega_{ik}/n$
   - $\hat{\Theta}_k = \arg\max_{\Theta} \left\{ \ln |\Theta_k| - tr(\tilde{S}_k \Theta_k) - \lambda_n ||\Theta_k||_1 \right\}$, where

$$\tilde{S}_k = \frac{\sum_{i=1}^{n} \omega_{ik} (\mathbf{y}_i \mathbf{y}_i^{'})}{\omega_{.k}}$$

and

$$\lambda_n = \frac{2\lambda}{\omega_{.k}}.$$

2.4 Simulation and Real-data Example

We generate data from two component mixtures and consider two different schemes based on λ. We study the consistency properties of the PMLE by allowing the sample size to grow. We subsequently applied our method to two real data "Mathematics scores" and "CellSignal" data.

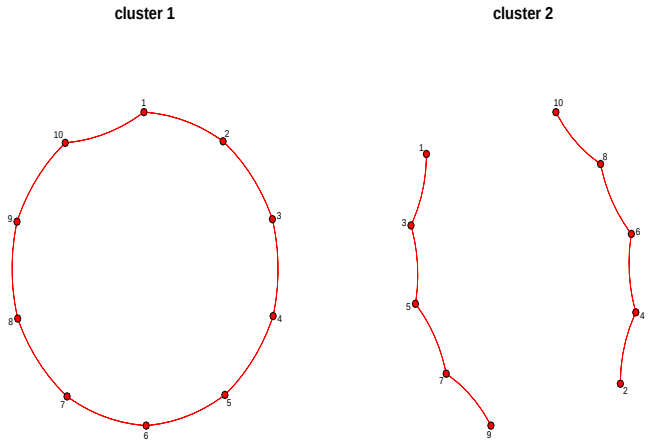

Figure 2.2. True graphical model of the 2 clusters

2.4.1 Simulation

We investigate the consistency properties of the PMLE using our penalized EM algorithm described in section 2.2. We simulate data $\mathbf{Y}_1, ..., \mathbf{Y}_n$ from two-component multivariate normal mixture models each with probability (true mixture proportion) equals 0.5 and inverse covariance matrix Θ_k built according to the following schemes.

$$\Theta_1(i,j) = \begin{cases} 1 \quad if \quad i = j \\ -0.4, \quad if|i-j| = 1 \\ 0, \text{elsewhere} \end{cases} \tag{2.28}$$

$$\Theta_2(i,j) = \begin{cases} 1 \quad if \quad i = j \\ -0.4, \quad if|i-j| = 2 \\ 0, \text{elsewhere} \end{cases} \tag{2.29}$$

Model	Bias(AD)/Frobenuis	F_1 score	TP	FP	Precison
n=100					
π	AD=0.1125				
Θ_1	F=1.7280	0.555	5	5	0.5
Θ_2	F=1.6221	0.529	9	15	0.375
n=300					
π	AD=0.067				
Θ_1	F= 0.9702	0.5333	8	14	0.3636
Θ_2	F= 0.8432	0.5882	10	14	0.4167
n=800					
π	AD=0.0625				
Θ_1	F=0.9279	0.5882	10	14	0.4166
Θ_2	F=0.4804	0.4705	8	18	0.3076
n=2000					
π	AD=0.0263				
Θ_1	F=0.4170	0.5925	8	11	0.4210
Θ_2	F=0.4465	0.625	10	12	0.4545
n=5000					
π	AD=0.002				
Θ_1	F=0.3529	0.6153	8	10	0.444
Θ_2	F=0.2883	0.6060	10	13	0.4347

Table I. The Absolute Deviation (AD), Frobenuis norm (F), the F_1 score, the True Positive (TP), the False Positive (FP) and the Precision of the PMLE for two-component mixture with $\lambda \propto \sqrt{n \log p}$.

The corresponding graphical model structures are depicted in Figure (2.2). For a fixed p, we consider two schemes one with $\lambda \propto \sqrt{n \log p}$ where $\lambda_n \propto \frac{1}{\sqrt{n}}$ and the other with $\lambda \propto \sqrt{\log p}$, where $\lambda_n \propto \frac{1}{n}$ each with increasing sample sizes, $n = (100, 300, 800, 2000, 5000)$ to examine the consistency of the PMLEs. In all cases, parameter estimation is achieved by maximizing the likelihood function via our penalized EM-algorithm. The results of our penalized EM-algorithm approach are compared based on the two different schemes corresponding to different values of λ.

Due to the effect of label switching, we are not able to assign correctly each parameter estimate to the right class. As a result, the estimates $\{(\pi_1, \Theta_1), (\pi_2, \Theta_2)\}$ will be interchangeably represented. We compute the Absolute

Deviation (AD) of the mixture proportions, and compare the Frobenuis norm of the difference between the true and estimated precision matrices for each cluster. In addition we compute the F_1 score, True positive (TP), False positive (FP), Precision and Recall for the PMLE.

Example 1. We considered the simulated two-component multivariate normal mixture models above and choose sequence of values of λ such that $c_1\sqrt{n\log p} \leq \lambda \leq c_2\sqrt{n\log p}$. On experimental basis we set $(c_1, c_2) = (0.1, 0.25)$. The performances of the penalized EM-algorithm corresponding to different sample sizes are presented in Table I.

The results show that as the sample size increases, the AD (for the mixture proportions) and the Frobenuis norms (for the precision matrices) decrease indicating the consistency of the PMLEs. At $n = 5000$, the AD for the mixture proportion is almost 0, indicating that our method has recovered precisely the true mixture distribution. We reported also the F_1 score, the True Positive (TP), the False Positive (FP), the Precision and the Recall of the PMLE. We recorded an overall improvement in the F_1 score as n increases.

Example 2. In this example, we again choose the same two-component multivariate Gaussian mixture models. In contrast to the model used in example 1, we have fixed the tuning parameter λ such that $c_1\sqrt{\log p} \leq \lambda \leq c_2\sqrt{\log p}$ and (c_1, c_2) remain unchanged. The performances of the penalized EM-algorithm corresponding to different sample sizes are presented in Table II. We again observe a decrease in both the Frobenuis norm and the AD as n increases even though we suffer from a deficiency in the AD of π for the case $n = 800$. However the AD is almost 0 at $n = 5000$. We note that this penalty decreases to 0 faster and as result tends to produce full graph as can be seen in the higher value recorded for false positive.

Comparing the 2 examples, we observe that the choice of λ plays a strong role in parameter and graph selection consistency of the resultant networks. The consistency properties of the PMLEs was achieved in both cases but our results indicate that the overall performance of the asymptotic behavior of $\lambda \propto \sqrt{n\log p}$ is more satisfactory. Even though both penalty decrease to 0 as n increases, $\lambda \propto \sqrt{n\log p}$ decreases slower resulting in a relatively sparser networks as compared to $\lambda \propto \sqrt{\log p}$.

2.4.2 Real-data Examples

Mathematics Scores Data

As a simple example of a data set to which mixture models may be applied, we consider the data set on marks in five mathematics exams score. This data set can be found in (Whittaker, 2009) and consists of 88 students who took examinations in

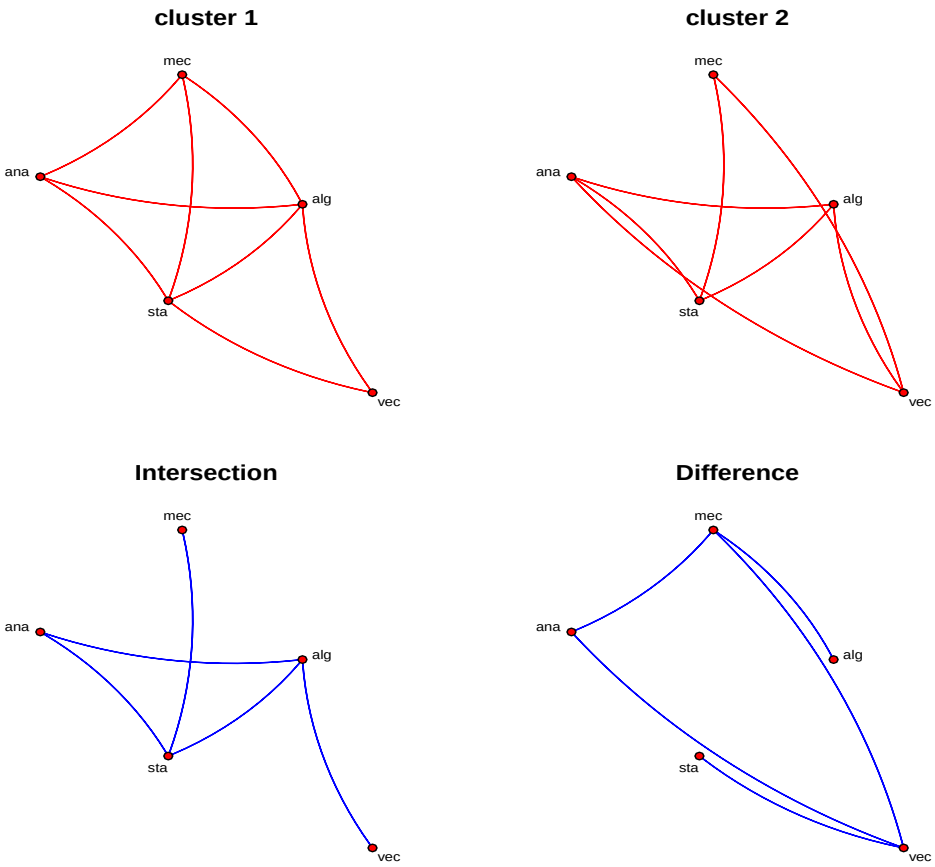

Figure 2.3. Graphical model of the 2 group of students

5 subjects namely mechanics, vectors, algebra, analysis, statistics. Some were with open book and others with closed book. Mechanics and vectors were with closed book.

We fit a two-mixture components to the data with a strong indication that there are two groups of students each with similar subjects interest. We applied our PMLE algorithm to the data with λ based on scheme 1. The pattern of interactions among the two groups were depicted in Figure (2.3). The network differences as well as similarities are also shown. The results indicate that 61% of students have similar subjects interest while 39% falls in other group of interest. In one group, we observe no interactions between mechanics and analysis nor statistics and vectors while in the other group such interactions do exist.

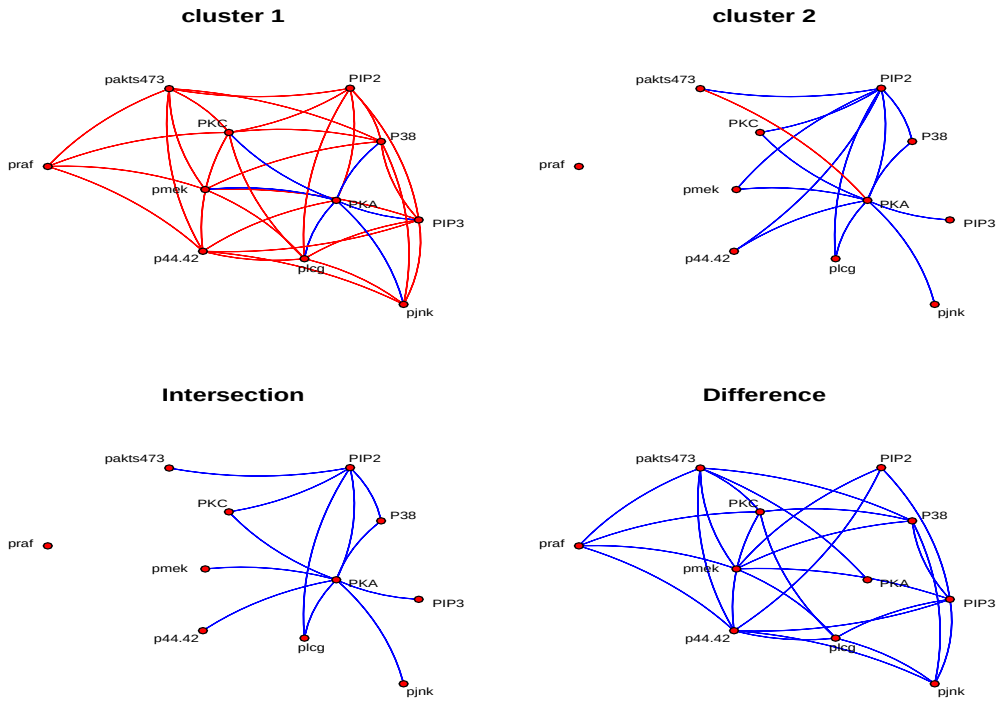

Figure 2.4. Graphical models of the CellSignal data with two mixtures of Gaussian distributions

Analysis of cell signalling data

We consider the application of our method on the flow cytometry dataset (cell signalling data) of Sachs et al. (2005). The data set contains flow cytometry of $p = 11$ proteins measured on $n = 7466$ cells. The CellSignal data were collected after a series of stimulatory cues and inhibitory interventions with cell reactions stopped at 15 minutes after stimulation by fixation, to profile the effects of each condition on the intracellular signaling networks. Each independent sample in the data set is made up of quantitative amounts of each of the 11 phosphorylated molecules, simultaneously measured from single cells.

We again fit a two-mixture component to the data. The result of applying our PMLE algorithm to the data set using the first scheme is shown Figure (2.4). The result indicates that 90% of the observation falls in one component whiles 10% falls in the other cluster. We also display the differences and similarities in the two components. The following proteins interaction were seen to be present in each of the

Model	Bias(AD)/Frobenuis	F_1 score	TP	FP	Precison
n=100					
π	AD=0.0307				
Θ_1	F= 3.4081	0.3446	10	32	0.2380
Θ_2	F= 3.4018	0.3181	7	29	0.1944
n=300					
π	AD=0.0356				
Θ_1	F=1.0539	0.3703	10	34	0.2272
Θ_2	F=0.8657	0.3137	8	35	0.1860
n=800					
π	AD=0.0669				
Θ_1	F=0.6419	0.3703	10	34	0.2272
Θ_2	F=0.7605	0.3018	8	37	0.1777
n=2000					
π	AD=0.0312				
Θ_1	F=0.5081	0.3168	8	34	0.1882
Θ_2	F=0.4150	0.3636	10	35	0.2222
n=5000					
π	AD=0.0065				
Θ_1	F=0.2771	0.3703	10	34	0.2272
Θ_2	F=0.2857	0.2692	7	37	0.1590

Table II. The Bias(AD), Frobenuis norm (F), F_1 score, True Positive (TP), False Positive (FP) and the Precision of the PMLE for two-component mixture with $\lambda \propto \sqrt{\log p}$.

two components: $(pakts473, PIP2)$, $(PKC, PIP2)$, $(PKA, pjnk)$, $(pmek, PKA)$ to mention but few. Differences in the interaction occur among the following proteins: $(pakts473, praf)$, $(PIP2, p44.42)$, $(PKC, plog)$; see Figure (2.4) for details.

2.5 Conclusion

We have developed a penalized likelihood estimator for Gaussian graphical mixture models. We have imposed an L_1 penalty on the precision matrix with extra condition preventing the likelihood not to degenerate. The estimates were efficiently computed through a penalized version of the EM-algorithm. By taking advantage of the recent development in Gaussian graphical models, we have implemented our method with

the use of the graphical lasso algorithm. We have provided consistency properties for the penalized maximum likelihood estimator in Gaussian graphical mixture model. Our results indicate a better performance in parameter consistency as well as in graph selection consistency for $\lambda = O(\sqrt{n \log p})$ or $\lambda_n \propto \frac{1}{\sqrt{n}}$. Our method is suitable for large networks recovering from non homogeneous data. Another interesting situation is when K, the number of mixture components in the model is unknown. This is a more practical problem than the one we have discussed and probably involves simultaneous model selection. This was implemented in our package `glassomix` as shown in the appendix.

Chapter 3

SSM of dynamic genetic networks

3.1 Introduction

Since the turn of the century a new scientific field has started to emerge: system biology has been brought to the fore front of life-science based research and development, (Bernhard, 2011). It is a biology-based, but inter-disciplinary field that focuses on the systematic study of complex interactions in biological systems. The aim of this holistic approach is to discover new emergent properties that may arise from the systemic view, which would not arise from reductionist approaches. The concept of gene networks is central in system biology. We view networks as comprising of nodes (the genes) and the links (chemical reactions) between them. They describe the idea of the stability and interconnectedness of molecular reactions. The challenge is to give these a precise statistical interpretation. In recent times, expression level of many of genes can be measured simultaneously through many techniques including DNA hybridization arrays (Wen et al., 1998; Derisi et al., 1997). A major challenge in system biology is to uncover, from such measurements, gene-protein interactions and key biological features of cellular systems.

We present a statistical method that infers the complexity, the dependence structure of the networks topology and the functional relationship between the genes; we also deduce the kinetic structure of the network. Our approach is based on the linear Gaussian state space models (SSM) of Fahrmeir and Kunstler (2009); Fahrmeir and Wagenpfeil (1997) or by Zoubin (2001); Yamaguchi et al. (2007) applied to real experimental data obtained from a well established model of T-cell activation, where relevant genes are monitored across various time points. Most publications only consider static Bayesian networks (Nir et al., 2000), that model discretized data but incorporate hidden variables. However there has been an increasing need for dynamic modelling that assumes the observed gene expression in the form of mRNA

to be continuous time series gene expression data and at the same time incorporate unknown factors such as hidden variables. We build a dynamic model of observed variables (RNA transcripts) and unobserved quantities commonly unmeasured protein regulators, and the relationships between the hidden state variables and the observed RNA transcripts. We infer the model structure as a biological network by estimating model interactions parameters through the EM algorithm (Dempster et al., 1977; Beal et al., 2005; Ghahramani and Hinton, 1996) combined with the Kalman smoothing algorithm (Shumway and Stoffer, 2005; Meinhold and Singpurwalla, 1983) in the context of maximum likelihood estimation. We use the bootstrap approach in (Efron, 1979) to infer the complex transcriptional response of the networks and to reveal interactions between components.

Choosing SSM to model networks kinetics has a number of advantages. Most importantly, it allows the inclusion of hidden regulators which can either be unobserved gene expression values or transcription factors (TFs). It can be used to model gene-gene and gene-protein interactions, represented by the matrices B and A respectively from Equation (1.13) . The hidden variables also allow us to handle noisy continuous measurements which represent the observed gene expression level at each time point. Next, the parameter estimates obtained through the EM algorithm and the state estimates from the Kalman filter have been shown to be consistent and asymptotically normal under some general conditions; (Ljung and Caines, 1979; Dent and Min, 1978). The EM algorithm itself guarantees at least a monotonically increasing likelihood.

Model selection or determining a suitable dimension of the hidden state is an additional complication. Rangel et al. (2004) approached the problem of deciding on a suitable dimension of the hidden state through cross validation. In their approach, they continuously increased the dimension of the hidden states and monitored the predictive likelihood using the test data; one major drawback of this approach is that it is very slow.

Several authors have exploited Kalman filtering and SSM of gene expression and used them to reverse engineer transcriptional networks. To this effect, Fang-Xiang et al. (2004), in modelling gene regulatory networks, used a two-step approach. In the first step, factor analysis is employed to estimate the state vector and the design matrix; the optimum dimension of the state vector k was determined by minimum BIC. In the second step, the matrix representing protein-protein translation is estimated using least squares regression. Rangel et al. (2004) have applied SSM to T-cell activation data in which a bootstrap procedure was used to derive a classical confidence interval for parameters representing gene-gene interaction through a re-sampling technique. Beal et al. (2005) approached the problem of inferring the model struc-

tures of the SSM using variational approximations in the Bayesian context through which a variational Bayesian treatment provides a novel way to learn model structure and to identify optimal dimensionality of the model. Recently, Bremer and Doerge (2009) used SSM to rank observed genes in gene expression time series experiments according to their degree of regulation in a biological process. Their technique is based on Kalman smoothing and maximum likelihood estimation techniques to derive optimal estimates of the model parameters; however, little attention has been paid to the dimension of the hidden state.

In this chapter we demonstrate how the EM algorithm with the Kalman smoothing algorithm is used in the maximum likelihood set-up to reverse engineer transcriptional networks from gene expression profiling data. By so doing we are able to add some useful interpretations to the model. We use the minimum AIC to determine the hidden state's optimal dimension.

The rest of the chapter is organized as follows. In section 3.2, we introduce the model, and give it a precise mathematical interpretation. Section 3.3 describes the inference method including the model selection procedure. Identifiability is also discussed briefly and we point out that if we simply estimate parameters of SSM without further constraints on parameter space, the parameters are not identifiable and the EM algorithm may get stuck to a local maximum. We assess in section 3.4 via simulation the performance of our method extensively in terms of F_1-score and false positive rates under various scenarios. Section 3.5 is the application of our model to real data (T-cell data) where we identify the network kinetics, by identifying genetic regulatory networks. We also summarize our results, analyze their statistical significance and their biological plausibility. We conclude with a discussion of the method used, possible extension, and a summary of related work in section 3.6.

3.2 State space model

Linear Gaussian state space models, also known as linear dynamical systems or Kalman filter models (Brown and Hwang, 1997; Dewey and Galas, 2000), are a class of dynamic Bayesian networks that relate temporary observation measurements y_t to some hidden state variable θ_t. We consider a sequence $(y_1, ..., y_T)$ of p-dimensional real-valued observation vectors through time, which we shall simply denote by $y_{1:T}$, representing a gene expression data matrix with p rows and T columns, where p and T are the number of genes and the measuring time points, respectively. The model assumes that the evolution of the hidden variables θ_t is governed by the state dynamics, which follows a first-order Markov process and is further corrupted by a Gaussian intrinsic biological noise η_t. However, these hidden variables are not

directly accessible but rather can be inferred through the observed data vector, y_t, namely the quantity of mRNA produced by the gene at time t. The observation y_t is a possibly time-dependent linear transformation of a k dimensional real-valued θ_t with observational Gaussian noise ξ_t. The model is given by assuming n_R biological replicates as follows:

$$\begin{cases} \theta_{tr} = Fx_{t-1,r} + Ay_{t-1,r} + \eta_{tr} \\ y_{tr} = Zx_{t,r} + By_{t-1,r} + \xi_{tr} \end{cases} \tag{3.1}$$

where $r = \{1, 2, ..., n_R\}$, F, A, Z and B represent the model interactions parameters of dimensions compatible with the matrix operations required in Equation(3.1). The terms η_t and ξ_t are zero-mean independent system noise and measurement noise, respectively with

$$E(\eta_t \eta_t^{'}) = Q, \qquad E(\xi_t \xi_t^{'}) = R \tag{3.2}$$

Both Q and R are assumed to be diagonal in many practical applications. The initial state x_0 is independently Gaussian distributed with mean $a_0 = 0$ and covariance Q_0. This model is more complex and represents an extension of the standard SSM described in chapter (1) as it includes various forms of feedback and can also be extended to include additional covariates.

A mathematical representation of the model is depicted in Figure (3.1) indicating two dynamics, the state and the observed, across 3 consecutive time points, where we assumed $k = p = 2$. We now collect the model interaction parameters into a single vector φ i.e $\varphi = \{G, Q, R, Q_0\}$ where $G = \begin{bmatrix} B & Z \\ A & F \end{bmatrix}$ is interpreted as a directed and weighted adjacency matrix of the graph of interactions.

3.3 Inference

3.3.1 Identifiability issues

Briefly speaking, a parameter of a dynamic system is said to be identifiable given some data if only one value of this parameter maximizes the observed likelihood. The identifiability property is important because it guarantees that the model parameter can be determined uniquely from the available data. Poor identifiability issues of the SSM stems from the fact that given the original model (Equation 3.1), and with the linear transformation of the state vector $x_t^* = Tx_t$, where T is a non-singular square matrix, we can find a different set of parameter vectors

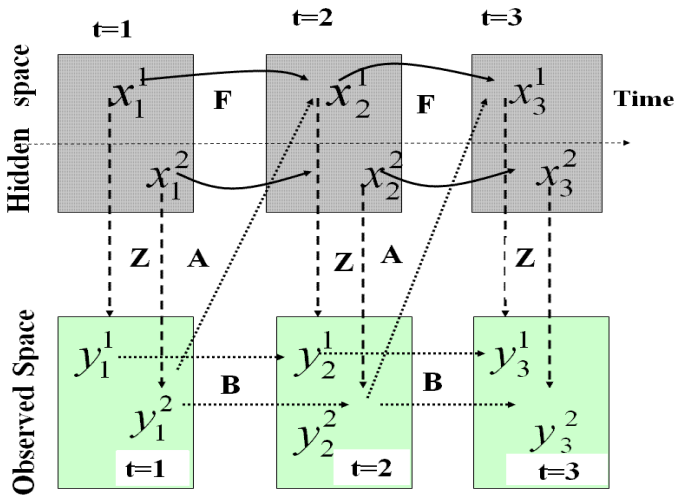

Figure 3.1. Two genes network representing an input-dependent SSM for Gene regulation with the vector of observed gene expression (y_t) and the hidden regulators of gene expression (x_t) at 3 different time points, where F, A, Z, and B correspond to the matrices in Equation (3.1).

$$\hat{\varphi}^* = \left\{\hat{G}^*, \hat{Q}^*, \hat{R}^*\right\}$$

that give rise to the same observation sequence $\{y_t, t = 1, 2, ..., T\}$ having the same likelihood as the one generated by the parameter vector φ. Hence, if we place no constraints on F, A, Z, B and possibly Q and R, there exist an infinite space of equivalent solutions $\hat{\varphi}$ all with the same likelihood value. To overcome such identifiability issues, further restrictions have to be imposed on the model. In our work, we assume Q and Q_0 to be identity matrices and R is set to be diagonal matrix. Subjecting Q to be identity only affects the scale of x and matrices A and Z.

We further assume that the errors $\{\eta_t, t = 1, ..., T\}$ and $\{\xi_t, t = 1, ..., T\}$ are jointly normal and uncorrelated. Also the number of time points or biological observations in microarray data are typically much smaller than the number of genes. This fundamental problem of high-dimensional statistical modelling of micro array data demands some care in the estimation of model parameters in the state space model. This problem is avoided by making sure that the number of observations exceed the total number of parameters to be estimated.

$$pTn_R > p^2 + 2kp + k^2. \tag{3.3}$$

This further puts the following bound on the dimension of the hidden states as given in Equation (3.4)

$$0 \leq k < -p + \sqrt{pTn_R}. \tag{3.4}$$

3.3.2 The likelihood function

We now restrict the model interaction parameters into the single vector $\varphi = \{F, A, Z, B, R\}$. As can be seen from Figure (3.1), the observations at time t, y_{tr} are conditioned on the past observations, $y_{(t-1)r}$ and on the regulators x_{tr} and also to infer for instance x_{tr}, we need $x_{(t-1)r}$ and $y_{(t-1)r}$. To that effect, under the Gaussian assumption we have the following:

$$\begin{aligned} x_{0r} &\sim \psi_k(0, I) \\ x_{tr}|x_{(t-1)r}, y_{(t-1)r} &\sim \psi_k(\tilde{x}_{tr}, I) \\ y_{tr}|x_{tr}, y_{(t-1)r} &\sim \psi_p(\tilde{y}_{tr}, R). \end{aligned}$$

where

$$\tilde{x}_t = Fx_{t-1} + Ay_{t-1},$$
$$\tilde{y}_t = Zx_t + By_{t-1},$$

and $\psi(\mu, \Sigma)$ is the normal density with mean μ and variance covariance Σ.

We now write the marginal likelihood function $l_y^m(\varphi)$ of the data y. This is given by

$$\begin{aligned} l_y^m(\varphi) &= \int \prod_{t=1}^{T} P(x_t|F, A, x_{t-1}, y_{t-1}) \times P(y_t|B, Z, x_t, y_{t-1}) dx \\ &= \int \prod_{t=1}^{T} \psi(x_t|\tilde{x}_t, \sigma_\eta^2 I)\psi(y_t|\tilde{y}_t, \sigma_\xi^2 I) dx. \end{aligned} \tag{3.5}$$

The full log-likelihood function of the complete data (y_{tr}, x_{tr}) denoted by $l_{y,x}(\varphi)$ is for simplicity given by

$$l_{y,x}(F, A, Z, B) = \sum_{r=1}^{n_R} l_{y_r x_r}^r(F, A, Z, B) \tag{3.6}$$

where $l^r_{y_r x_r}(F, A, Z, B)$ is the complete log-likelihood of the r^{th} replicate and is given by

$$\begin{aligned} l^r_{y_r x_r}(F, A, Z, B) &= \sum_{t=1}^{T} l_{y_t|x_t, y_{(t-1)}}(Z, B) + \sum_{t=1}^{T} l_{x_t|x_{(t-1)}, y_{(t-1)}}(F, A) \\ &= -\frac{1}{2\sigma^2_\xi} \sum_{t=1}^{T} (y_t - \tilde{y_t})' (y_t - \tilde{y_t}) - \frac{T}{2}\log(\sigma^2_\xi) \\ &- \frac{1}{2\sigma^2_\eta} \sum_{t=1}^{T} (x_t - \tilde{x_t})' (x_t - \tilde{x_t}) - \frac{T-1}{2}\log(\sigma^2_\eta) \end{aligned} \tag{3.7}$$

ignoring constant term.

3.3.3 Joint parameter estimation via EM algorithm

Our aim is to estimate the model parameter φ which (excluding R) indicates connectivity matrix of the directed genomic graph that maximizes the marginal likelihood function $l^m_y(\varphi)$ given in Equation (3.5). The integral in Equation (3.5) is difficult because of the presence of the hidden variables x. For that matter we use the EM algorithm to learn the parameters of the model. The idea stems from the fact that if we did have the complete data (y_t, x_t) it will be straight forward to obtain MLEs of φ using multivariate normal theory. In this case, we do not have the complete data and the EM algorithm gives us an iterative method for finding the MLE of φ using the observed data y_t, by successively maximizing the conditional expectation of the complete data likelihood given the observed values. It is only when we are able to estimate the parameter φ that we can expect to obtain some useful interpretations of the biological system networks.

The EM-algorithm for SSM was formulated by Shumway and Stoffer (1982) and Shumway (2000). To this effect the algorithm requires the computation of the conditional expectation of the log-likelihood given the complete data. The algorithm is a two-stage procedure in which we begin with a set of trial initial values for the model parameter to calculate the Kalman smoother. The Kalman smoother is then input into the M-step to update parameter estimates. The algorithm alternates recursively between an expectation step followed by a maximization step.

The expected log-likelihood function: The E-step

This step of the EM algorithm involves the calculation of the first two moments x_t of the hidden states. Let $\mathbf{Q}$ denote the expected log-likelihood. Then from Equation 3.6, $\mathbf{Q}$ becomes

$$\begin{aligned}
\mathbf{Q}(\varphi|\varphi^*) &= E_x\left[l_{y,x}(\varphi)|y,\varphi^*\right] \\
&= \sum_{r=1}^{n_R} E_x\left[l^r_{y_r,x_r}(\varphi)|\varphi^*,y\right] \\
&= \sum_{r=1}^{n_R} E_x\left[l^r_{y_r,x_r}(Z,B)|y,\varphi^*\right] + \sum_{r=1}^{n_R} E_x\left[l^r_{y_r,x_r}(F,A)|\varphi^*,y\right]. \\
&= \mathbf{Q}(Z,B) + \mathbf{Q}(A,F).
\end{aligned} \tag{3.8}$$

$\varphi^* = (Z^*, B^*, F^*, A^*)$ is the estimate obtained from the previous M-step

The calculation of $\mathbf{Q}(\varphi|\varphi^*)$ in Equation (3.8) involves finding $E(x)$ and $E(x^{'}x)$ for each replicate r; these forms are supplied by the Kalman smoothing algorithm. The above implies that for each replicate we run the Kalman smoothing algorithm to find the expected hidden states and their variance-covariance components and these are joined together to get $\mathbf{Q}(\varphi|\varphi^*)$. In essence the hidden state is estimated every time the algorithm visits the E-step through the Kalman filter and smoother.

The update equations: The M-step.

We then move to the M-step where given the already estimated hidden state, we update the model interaction parameters. A new parameter set φ^{i+1} is computed by estimating the parameters that maximize Equation (3.8); the expected log-likelihood function that is

$$\hat{\varphi} = \arg\max_{\varphi}\left\{\mathbf{Q}(\varphi|\varphi^*)\right\} \tag{3.9}$$

These can be solved in closed form in the following manner.

$$\frac{\partial}{\partial\varphi}\mathbf{Q} = 0$$

and then solve for the parameter value that sets the partial derivative to zero. It is also important to note that the partial derivatives are taken with respect to matrices F, A, Z and B.

The update estimates for matrix Z and B.
Take the derivative of $\mathbf{Q}$ with respect to Z and B and equating them to 0. We write $\mathbf{Q}(Z,B)$ as

$$\begin{aligned}
\mathbf{Q}(Z,B) &= \sum_{r=1}^{n_R} E_{x,\varphi^*}\left[l^r_{y_r,x_r}(Z,B)\right] \\
&= -\sum_{r=1}^{n_R}\sum_{t=1}^{T} y_{tr}^{'} y_{tr} + 2\sum_{r=1}^{n_R}\sum_{t} E(x_{tr}^{'} Z y_{tr}) \\
&\quad +2\sum_{r=1}^{n_R}\sum_{t} y_{(t-1)r}^{'} B^{'} y_{tr} - \sum_{r=1}^{n_R}\sum_{t}^{T} Z E(x_{tr}^{'} x_{tr} Z^{'}) \\
&\quad -2\sum_{r=1}^{n_R}\sum_{t}^{T} E(x_{tr}^{'} Z^{'} B y_{(t-1)r}) - \sum_{r=1}^{n_R}\sum_{t}^{T} B^{'} y_{(t-1)r}^{'} y_{(t-1)r} B
\end{aligned}$$

Setting $\frac{\partial}{\partial Z}\mathbf{Q}(Z,B)$ and $\frac{\partial}{\partial B}\mathbf{Q}(Z,B)$ equal 0 result in two linear systems of equations in the form:

$$\begin{aligned}
0 &= -\frac{1}{2\sigma^2_{\xi_{tr}}}\sum_{r=1}^{n_R}\sum_{t=1}^{T}[-2y_{tr}E(x_{tr}^{'}) \\
&\quad +2\hat{Z}E(x_{tr}x_{tr}^{'}) + 2\hat{B}y_{(t-1)r}E(x_{tr}^{'})]
\end{aligned} \tag{3.10}$$

and

$$\begin{aligned}
0 = \; - \; & \frac{1}{2\sigma^2_{\xi_{tr}}}\sum_{r=1}^{n_R}\sum_{t=1}^{T}[-2y_{(t-1)r}y_{tr}^{'} + 2y_{(t-1)r}E(x_{tr}^{'})\hat{Z}^{'} \\
& +2(y_{(t-1)r}y_{(t-1)r}^{'}\hat{B}^{'})]
\end{aligned} \tag{3.11}$$

Equations (3.10) and (3.11) could also be re-written as

$$-M_{yx} + \hat{Z}M_{xx} + \hat{B}M_{L(y)x} = 0 \tag{3.12}$$

$$-M_{L(y)y} + M_{L(y)x}\hat{Z}^{'} + M_{L(y)L(y)}\hat{B}^{'} = 0 \tag{3.13}$$

where

$$M_{yx} = \sum_{rt} y_{tr}E(x_{tr}^{'})$$

$$M_{xx} = \sum_{rt} E(x_{tr} x_{tr}^{'})$$

$$M_{L(y)x} = \sum_{rt} y_{(t-1)r} E(x_{tr}^{'})$$

$$M_{L(y)y} = \sum_{rt} y_{(t-1)r} y_{tr}^{'}$$

$$M_{L(y)L(y)} = \sum_{rt} y_{(t-1)r} y_{(t-1)r}^{'}$$

and $L(y)$ in Equations (3.12) and (3.13) is the shift operator on matrix y. From Equation (3.12),

$$\hat{Z} = M_{yx} M_{xx}^{-1} - \hat{B} M_{L(y)x} M_{xx}^{-1} \tag{3.14}$$

Substitute Equation (3.14) into Equation (3.13), gives

$$\begin{aligned} \hat{B} M_{L(y)L(y)} &= M_{yL(y)} - M_{yx} M_{xx}^{-1} M_{xL(y)} \\ &+ \hat{B} M_{L(y)x} M_{xx}^{-1} M_{xL(y)} \end{aligned} \tag{3.15}$$

Therefore

$$\begin{aligned} \hat{B} &= \left[M_{yL(y)} - M_{yx} M_{xx}^{-1} M_{xL(y)} \right] \\ &\times \left[M_{L(y)L(y)} - M_{L(y)x} M_{xx}^{-1} M_{xL(y)} \right]^{-1} \end{aligned} \tag{3.16}$$

From Equation (3.13)

$$\hat{B} M_{L(y)L(y)} = M_{yL(y)} - \hat{Z} M_{xL(y)} \tag{3.17}$$

This implies

$$\hat{B} = M_{yL(y)} M_{yL(y)}^{-1} - \hat{Z} M_{xL(y)} M_{yL(y)}^{-1} \tag{3.18}$$

Substitute Equation (3.18) into Equation (3.12), gives

$$\begin{aligned} \hat{Z} M_{xx} &= M_{yx} - M_{yL(y)} M_{L(y)L(y)}^{-1} M_{L(y)x} \\ &+ \hat{Z} M_{xL(y)} M_{L(y)L(y)}^{-1} M_{L(y)x} \end{aligned} \tag{3.19}$$

Rearranging Equation (3.19), we have

$$\begin{aligned}\hat{Z} &= \left[M_{yx} - M_{yL(y)}M^{-1}_{L(y)L(y)}M_{L(y)x}\right] \\ &\times \left[M_{xx} - M_{xL(y)}M^{-1}_{L(y)L(y)}M_{L(y)x}\right]^{-1} \end{aligned} \tag{3.20}$$

Equations (3.16) and (3.20) are the update equations in the maximization step used to infer the parameters in the observation dynamics.

In the same manner we derive the updates equations for A and F for the model interaction parameters in the state dynamics model.

We write $\mathbf{Q}(A, F)$ as

$$\begin{aligned}\mathbf{Q}(A,F) &= \sum_{r=1}^{n_R} E_{x,\varphi^*}\left[l^r_{y_r,x_r}(F,A)\right] \\ &= C_2 + 2\sum_{t=1}^{T} y^{'}_{t-1}A^{'}E(x_t) + 2\sum_{t-1}^{T} E(x^{'}_t)FE(x_{t-1}) \\ &- \sum_{t=1}^{T} F^{'}Ex^{'}_{t-1}x_{t-1}F - 2\sum_{t=1}^{T} y^{'}_{t-1}A^{'}FEx_{t-1} \\ &- \sum_{t=1}^{T} A^{'}y^{'}_{t-1}y_{t-1}A \end{aligned}$$

where C_2 is a constant.

Setting $\frac{\partial}{\partial F}\mathbf{Q}(F, A)$ and $\frac{\partial}{\partial A}\mathbf{Q}(F, A)$ equal 0 result in two linear system of equations in the form:

$$\sum_{rt} E(x^{'}_{t-1}x^{'}_t) - \sum_{rt} E(x_{t-1}x^{'}_{t-1})\hat{F}^{'} - \sum_{rt} E(x_{t-1}y^{'}_{t-1})\hat{A}^{'} = 0 \tag{3.21}$$

and

$$\sum_{rt} y_{t-1}E(x^{'}_t) - \sum_{rt} y_{t-1}E(x^{'}_{t-1})\hat{F}^{'} - \sum_{rt} y_{t-1}y^{'}_{t-1}\hat{A}^{'} = 0 \tag{3.22}$$

Equations (3.21) and (3.22) could also be re-written as

$$M_{L(x)x} - M_{L(x)L(x)}\hat{F}^{'} - M_{L(x)L(y)}\hat{A}^{'} = 0 \tag{3.23}$$

$$M_{L(y)x} - M_{L(y)L(x)}\hat{F}^{'} - M_{L(y)L(y)}\hat{A}^{'} = 0 \tag{3.24}$$

From Equation (3.23), we have

$$\hat{F} = M_{xL(x)}M^{-1}_{L(x)L(x)} - \hat{A}M_{L(y)L(x)}M^{-1}_{L(x)L(x)} \tag{3.25}$$

Substitute Equation (3.25) into Equation (3.24), gives

$$\begin{aligned} \hat{A}M_{L(y)L(y)} &= M_{xL(y)} - M_{xL(x)}M^{-1}_{L(x)L(x)}M_{L(x)L(y)} \\ &+ \hat{A}M_{L(y)L(x)}M^{-1}_{L(x)L(x)}M_{L(x)L(y)} \end{aligned} \tag{3.26}$$

Therefore

$$\begin{aligned} \hat{A} &= \left[M_{xL(y)} - M_{xL(x)}M^{-1}_{L(x)L(x)}M_{L(x)L(y)}\right] \\ &\times \left[M_{L(y)L(y)} - M_{L(y)L(x)}M^{-1}_{L(x)L(x)}M_{L(x)L(y)}\right]^{-1} \end{aligned} \tag{3.27}$$

Next, from Equation (3.24), we have

$$\hat{A} = M_{xL(y)}M^{-1}_{L(y)L(y)} - \hat{F}M_{L(x)L(y)}M^{-1}_{L(y)L(y)} \tag{3.28}$$

Substitute Equation (3.28) into Equation (3.23), gives

$$\begin{aligned} \hat{F}M_{L(x)L(x)} &= M_{xL(x)} - M_{xL(y)}M^{-1}_{L(y)L(y)}M_{L(y)L(x)} \\ &+ \hat{F}M_{L(x)L(y)}M^{-1}_{L(y)L(y)}M_{L(y)L(x)} \end{aligned} \tag{3.29}$$

Rearranging Equation (3.29) gives

$$\begin{aligned} \hat{F} &= \left[M_{xL(x)} - M_{xL(y)}M^{-1}_{L(y)L(y)}M_{L(y)L(x)}\right] \\ &\times \left[M_{L(x)L(x)} - M_{L(x)L(y)}M^{-1}_{L(y)L(y)}M_{L(y)L(x)}\right]^{-1} \end{aligned} \tag{3.30}$$

Equations (3.27) and (3.30) are the update equations in the maximization step used to infer the parameters in the state dynamics.

The entire EM algorithm can be regarded as alternating between Kalman filtering and smoothing recursions and the normal maximum likelihood estimators as given in the update equations.

3.3.4 Choice of hidden state dimension: AIC_c

Model selection or the determination of the optimum dimension of the hidden state k is important to the application of SSM to network reconstruction. Popular model selection criteria include Akaike's Information Criterion (AIC) (Akaike, 1974) and the Bayesian Information Criterion (BIC) (Schwarz, 1978). We apply Akaike's Information Criterion (AIC) method for our model selection. AIC is aimed at finding the best approximating model to the unknown data generating process via minimizing the estimated expected K-L divergence, i.e. AIC's try to find the best approximation among the models we actually look at. Given the log-likelihood function l, AIC for a model with k-dimensional state vector is given by:

$$AIC(k) = -2l(y_t|\hat{\varphi}_k) + 2P \tag{3.31}$$

with P the number of estimated parameters, and $l(y_t|\hat{\varphi}^k)$ the log-likelihood of the observed data. As recommended by Burnham and Anderson (2002), we have applied AIC_c (AIC with a correction for finite sample size) for our model selection procedure. The reason being that AIC_c estimates the expected discrepancy with less bias than AIC. [1] The AIC_c is given by

$$AIC_c(k) = -2l(y_t) + 2P\left[\frac{N}{N-P-1}\right] \tag{3.32}$$

where $N = pTn_R$ represents total number of observations and $P = p^2+2kp+k^2$ is the total number of estimated parameters and we settle on the hidden state dimension that has the minimum AIC_c, i.e we find k such that

$$\hat{k} = \arg\min_k \{AIC_c(k)\}. \tag{3.33}$$

In this case, we successively increase the number of hidden states and monitor the behavior of AICc i.e., for each run of the EM algorithm, we increase k.

3.3.5 Network Reconstruction by Bootstrapping

In our procedure, we use a bootstrap approach to find confidence intervals for the parameters defining our model. By so doing we compute the bootstrap distribution of the estimator of φ.

[1] In the framework of normal linear regression models (both univariate and multivariate), the penalty term of AICc provides an exact expression for the bias adjustment. .

Let $\hat{\varphi}$ denote the MLE of the parameters defining our model; $\hat{\varphi}$ are estimated using the EM algorithm described in previous section. Suppose that for all $r \in \{1, ..., n_R\}$, y_r where $y_r \in \mathbb{R}^{P \times T}$ represents the data. The bootstrap procedure adopted is outlined below:

1. Obtained $\tilde{y}_1, ...\tilde{y}_{n_R}$. This is the model fit, an output of the Kalman filter.
2. Calculate, for all r in $\{1, ..., n_R\}$, the innovation errors $\xi_r = y_r - \tilde{y}_r$.
3. Sample with replacement from ξ_r to obtain ξ_r^*.
4. For all r in $\{1, ..., n_R\}$, make new data y_r^* through $y_r^* = y_r + \xi_r^*$.

Given each new data we estimate among other things, the bootstrap set of parameters $\left\{\hat{\varphi_b^*}; b = 1, ..., N_b\right\}$ through the EM algorithm. Stated differently for each bootstrap data, the parameters that maximize the likelihood of the bootstrap data are found, and then obtain the sampling distributions of the estimators of the elements of φ. The results of the bootstrapping are the distribution of the parameters and we proceed to make statistical inferences about those underlying parameters by computing confidence interval for each of them; (Wild et al., 2004; Shumway and Stoffer, 2005)

3.4 Simulation studies

In order to evaluate the performance of our method for analyzing gene expression data, we simulate artificial data and applied our proposed method to the simulated data according to the model described in Equation (3.1) with 10 time points, $p = 3$ as number of genes, and $k = 2$ TFs. The true newtork is depicted in Figure (3.2) (left).

In applying the EM- algorithm, parameters were initialized as follows: Z and F are assumed to be identity matrices whiles we initialize A to be zero. For B we perform a simple linear regression where we regress current genes on its previous ones and R assumes the usual variance estimate from the regression. We applied the bootstrap procedure to the data and identified the significant and non-significant parameters defining our model or identifying the dynamics of the networks. We achieved this by computing bootstrap confidence intervals on element φ_l of φ; it is clear that the confidence intervals will enable us to decide which elements φ_l will be set to zero and which will not. The analysis now turn to a decision problem where we formulate two hypotheses, namely,

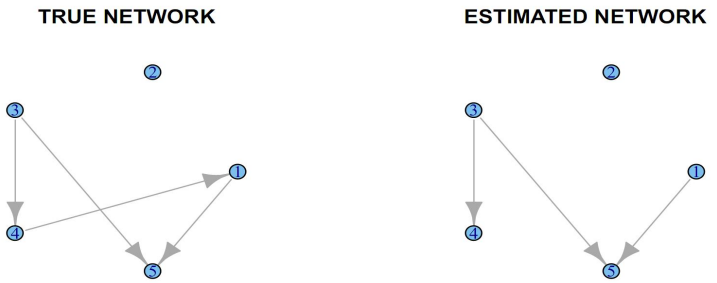

Figure 3.2. The true network G (left) and recovered network (right) $\hat{G}$

$$H_0 : \hat{\varphi}_l = 0$$

$$H_1 : \hat{\varphi}_l \neq 0$$

where rejecting H_0 indicates the presence of connection among the genes, meaning that the particular interaction in the matrix is considered to be statistically significant. With k equals 2, we obtained the upper and lower bounds of the confidence interval of the vectorized elements of the bootstrap estimated parameters, in the order of F_{ij}, A_{ij}, Z_{ij}, B_{ij}. The recovered network is shown in Figure (3.2) (right)

To provide a baseline through which we could evaluate the performance of the proposed method for gene regulatory network, we calculate the True Positive Rate (TPR), False Positive Rate (FPR) and the F_1 score of the matrix G representing the entire genomic interaction. Table (I) shows the simulation result at varying number of replicates. Perfectly recovering the network corresponds to a TPR of 1. According to Table (I), as the number of replicates n_R increases from 10 to 50, the TPR is as high as 75% whiles the FPR is as small as 19% with F_1 score of about 54%. This demonstrates the efficiency of our method.

n_R	10	25	40	50
TPR	0.75	0.75	0.75	0.75
FPR	0	0.19	0.19	0.19
F-score	0.85	0.54	0.54	0.54

Table I. Simulation result for TPR and FPR as the number of replicates n_R increases from 10 to 50.

3.5 Application

k	2	3	4	5	6	8
AICc	3386201	2537048	2524402	2849645	2800490	2884533

Table II. Estimates of AIC_c as a function of k.

For this study, to demonstrate the application of our reverse engineering method, we used publicly available data, the results of two experiments used to investigate the expression response of human T-cells to PMA and ionomicin treatment. The data is a combination of two data set namely tcell.34 and tcell.10. The first data set (tcell.34) contains the temporal expression levels of 58 genes for 10 unequally spaced time points. At each time point there are 34 separate measurements. The second data set (tcell.10) comes from a related experiment considering the same genes and identical time points, and contains 10 further measurements per time point. At each time point there are 44 separate measurements or replicates. It was assumed that the 44 replicates have a similar underlying distribution. Given that the t-cell experiment is a time course gene expression data with technical replicates we expect more reliable estimation and inference results by applying our method. Corresponding to each gene expression y_{tr}, we also generated technical replicates for the hidden variables x_{tr}. With $p = 58$ genes, $R = 44$ as replicates and $T = 10$, the constraint represented by Equation (3.3) is satisfied, indicating that we have enough data to estimate our parameters. The dimension of the hidden variables was determined using AICc as explained in section (3.3.4). Table (II) shows the behavior of AICc with corresponding k's. It turns out that $k = 4$ is the optimum number of the hidden states as compared to Rangel et al. (2004) and Beal et al. (2005) who obtained 9, 14 respectively under different criteria.

In essence, we treated the data as a time series measurement y_{t_r}, $t = 1, 2, ..., 10$ and $r = 1, 2, ..., 44$. For each replicate, y_t and x_t consist of 58 genes and 4 transcriptions factors respectively, each, measured at 10 different time points, i.e for each replicate r, y and x are of dimension (58×10), (4×10) respectively. Some of these genes include RB1, CCNG1, TRAF5, CLU.... The parameters Q and Q_0 were fixed.

We then applied the EM algorithm to the data and Figure (3.3) shows the estimated values of the hidden variables x i.e the expression pattern of the 4 latent variables across time. Based on the test, with 95% confidence level, we plot the connectivity matrix of the directed genomic network $\hat{G}$. The output is a directed graph

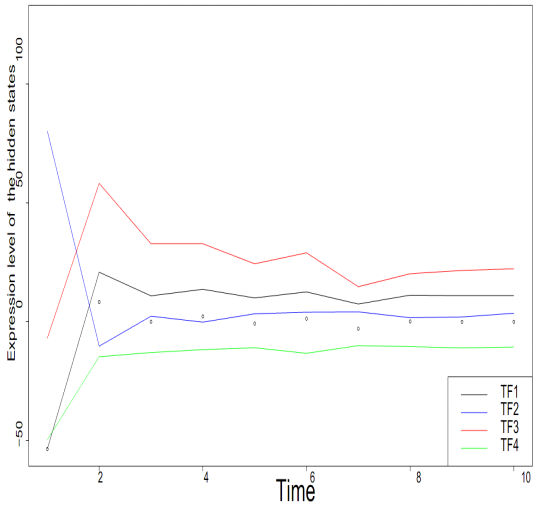

Figure 3.3. EM algorithm on the T-cell data showing the expression level of the latent variables across time

showing connections from one gene expression variable at a given time point t to another gene expression variable whose expression it influences at the next time point, $t+1$. The arrows indicate the direction of the regulation. The entire directed graph $\hat{G}$ gives 350 genomic interactions. Figure (3.4) represents a portion of the interaction network $\hat{\varphi}$ where we indicate genes that have at least 3 outwards connections. These genes include the FYN-binding protein gene FYB, the JUND proto-oncogene, the CD69 antigen p60, early T-cell activation antigen to mention but few. Figure (3.5) is the sub-network produced at 95% confidence level and it represents the interaction between, two Jun proteins family namely JUNB and JUND and various genes involved in programmed cell death. Our method through Figure (3.5) supports the anti-proliferation and anti-apoptotic role of JUND. We also recover the topology of the genes FYB through Figure (3.6). The structure of the network is visualized using the R package for Network analysis and visualization igraph.

According to our method, the following genes were mostly seen as regulatory genes. These genes includes the JUND proto-oncogene, the CLU gene, the cell division cycle 2 CDC2, the FYN-binding protein gene FYB, TRAF5, the CD69, and the GATA-binding protein 3. The latent variables were also seen to regulate the expression level of most genes as could be seen in Figure (3.4).

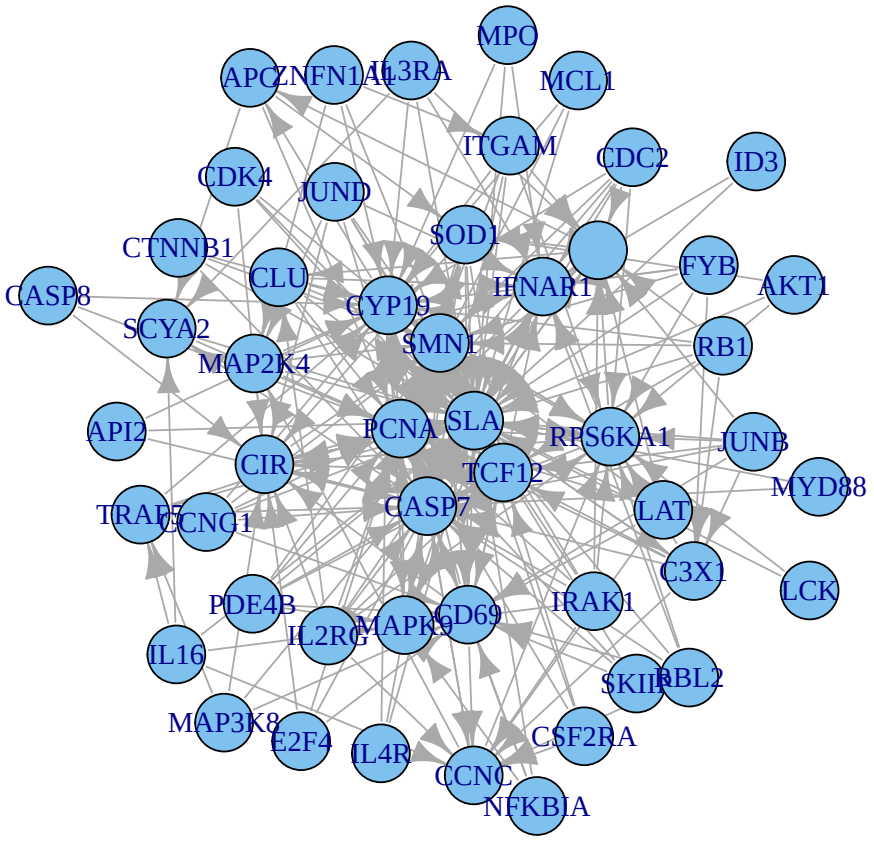

Figure 3.4. **Sub-network found representing the genomic interactions $\hat{G}$, of genes with at least 3 outwards connections, nodes refer to gene expressions in the form of proteins or RNAs; empty nodes refer to latent variables**

Our approach has revealed interesting features in the family of Jun genes. The

network in Figure (3.5) provides support for interesting biological properties some of which also confirmed in (Rangel et al., 2004) and (Beal et al., 2005); but we also found new connections. In our work, we found interactions between the proto-oncogene JUNB, the apoptosis-related cysteine protease genes CASP4 and CASP8. The implication is that JUNB is clearly modelled as a pro-apoptotic gene by activating CASP4 and CASP8. This interaction was also was recovered by Beal et al. (2005). We however found no interaction between JUNB and MAP3K8. Also Figure (3.5) reveals that the proto-oncogene JUND activates the GATA-binding protein 3 but represses the expression level of the cell division cycle 2 (CDC2). This further supports the anti-proliferative JUND. Furthermore, in our model, the survival of motor neuron 1 gene SMN1 and the cell division cycle CDC2 influence the expression level of JUNB and MAPK8 respectively. JUNB activates the expression level of CDC2. A critical comparison of our Figure (3.5) to that of similar sub-networks found in the work of Andrea et al. (2010) and Beal et al. (2005) shows that in all the 3 sub-networks, JUND regulates the expression level of CDC2. JUNB activates CASP8 in the sub-network found by Beal et al. (2005) and indirectly regulates CASP8 through CASP4 in the sub-networks found by Andrea et al. (2010). However we found interaction between JUNB and both CASP8 and CASP4.

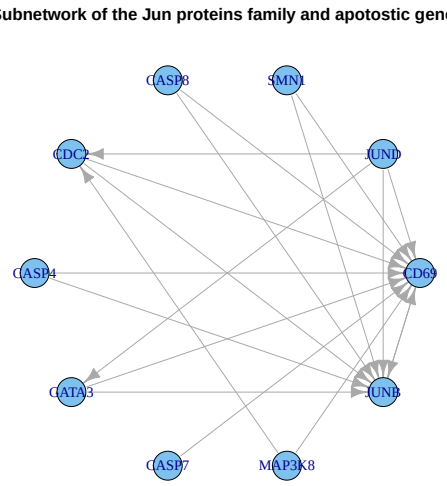

Figure 3.5. Sub-network found representing the interactions between Jun proteins family and apotostic genes

The gene FYN-binding protein FYB found to occupy one of the most crucial positions in the network recovered by Rangel et al. (2004) also has a high degree of connectivity in our work; and Figure (3.6) reveals some crucial genes that are found to be directly connected to FYB. Most importantly, in our model, FYB influences the expression level of genes such as the early T-cell activation marker CD69, the JUNB proto-oncogen. FYB is also seen to be connected to genes such as APC, API2, and CIR. Clearly, these results support the fact that FYB mRNA levels are predictive of the expression level of a number of genes.

The hidden state dimensionality was found to be 4, a result similar to the work of Andrea et al. (2010) in which they developed an iterative empirical Bayesian procedure with a Kalman filter to estimate the posterior distributions of network parameters. Rangel et al. (2004) found the dimension of the hidden state to be 9 through cross validation technique while Beal et al. (2005) through a variational Bayesian approach obtained 14. At 95% confidence level, we found no significant interactions among the hidden variables or transcription factors. However their role in the transcription process can not be ignored as the inferred matrix Z representing instantaneous protein-RNA transcription was not sparse signifying that the transcription factors regulate the expression level of most mRNAs.

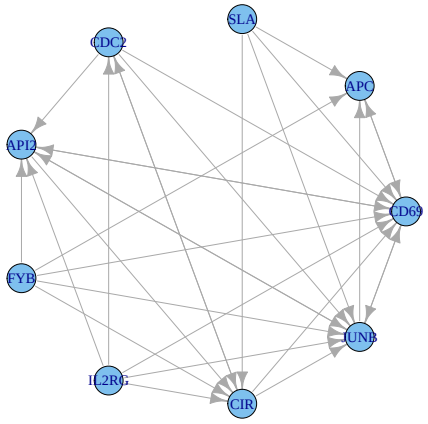

Figure 3.6. Sub-network found representing the topology of gene FYB in connection with other genes

3.6 Conclusion

In this chapter, we have developed a novel state space model in inferring regulatory networks from (high) dimensional data (e.g., gene expression) using linear Gaussian state space models. The EM algorithm was used to estimate the parameters because of hidden states and AICc criteria was used to select the number of hidden states. Parametric bootstrap was used to determine the selection of parameters. The proposed method offers significant advantages over other methods that have recently appeared in the literature. For example, Beal et al. (2005) used a variational Bayesian methodology which is an approximation of the posterior distribution of the parameters, while we did exact inference of the parameters. Rangel et al. (2004) used cross validation as model selection technique which is quite slow as compared to AIC. Bremer and Doerge (2009) used an *ad hoc* method for selecting the hidden state dimensionality k, while our method uses a data-driven approach. Also our model allows for dynamic correlation over time, as each observation and hidden state depend explicitly on some function of previous observations as opposed to the model described by Yamaguchi and Higuchi (2006); Perrin et al. (2003); Fang-Xiang et al. (2004). Their model does not allow for RNA-protein translation and RNA-RNA interactions through the matrix A and B respectively in our model.

One fundamental assumption in our proposed model is the first-order linear dynamics in the state and observation equations of the SSM. This assumption can only be an approximation to the true nature of a complex biological system since more realistic models of gene regulatory interactions surely include complex interactions or nonlinear relationships. Our linear dynamics assumption is a stepping stone upon which a future model with non-linear dynamics will be explored. With application to the t-cell data, We have discovered new interactions that do not find support in the current literature; as as part of our future work we will investigate these interactions further and possibly redefine our model.

Furthermore ML approach is prone to over-fitting, especially in high-dimensional statistical modelling. A natural way to avoid this over-fitting is through regularization. Also gene regulation tends to be sparse. In the next chapter we plan to employ a penalized maximum likelihood strategy in the context of the EM algorithm in the state space model.

Chapter 4

SSM with L_1 regularization constraint

4.1 Introduction

Reverse engineering transcriptional networks or modelling differential gene expressions as a function of time is providing a new insight for biological research. Technology is now available to track the expression pattern of thousands of genes in a cell in a regulated fashion and to trace the interactions of many of the products of these genes (Bower and Bolouri, 2001). However, the sheer dimensionality of all possible networks combined with the noisy nature of the observations and the complex structure of genomic regulation and signaling have meant that simply reading off a network from the data turned out somewhat optimistic. Instead, only statistical models of sufficient biological relevance are capable of discovering direct and indirect interactions between genes, proteins and metabolites. The last decade has seen an explosion of techniques to infer networks structure from microarray data. Models have now been developed to capture how information is stored in DNA, transcribed to mRNA, translated to proteins and then from protein structure to function. These models include Boolean networks based on Boolean logic (Kauffman, 1993; Patrik et al., 2000) where each gene is assumed to be in one of two states"expressed" or "not expressed", graphical Gaussian models (Schfer and Strimmer, 2005), Dynamic Bayesian Networks (Perrin et al., 2003), vector autoregressive models -VAR- (Fujita et al., 2007), ordinary differential equation models (Quach et al., 2007; Cao and Zhao, 2008) in which the state is a list of the concentrations of each chemical species and the concentrations are assumed to be continuous, stochastic differential equation models (Chen et al., 2005) and finally state space models (Rangel et al., 2004; Beal et al., 2005).

Integrating these models in mainstream statistics is an exciting challenge from a theoretical, computational, and applied perspective. Among the above mentioned

networks modelling techniques, ordinary differential equations -ODEs- have been established in recent years to model, gene regulatory or more generally, biochemical-networks, since they provide a detailed quantitative description of transcription regulatory networks. On the downside, they contain a large number of model parameters and are not well suited to deal with noisy data. Current methods for estimating parameters in ODEs from noisy data are very computationally intensive (Ramsay et al., 2007). Clearly, not all systems can be modeled with differential equations. Specifically, differential equations assume that changes of states are continuous and deterministic. Stochastic differential equations -SDE- is an alternative framework for genetic interactions inference where one writes the system of differential equations as before, then adds a noise term to each, but practically SDE is difficult to handle from both theoretically and computationally. It is especially difficult to infer the structure in high dimensional settings in both frameworks.

In this work, we consider a penalized state space model framework, a framework which consists of two different spaces, i.e a latent "protein" space and an observed "mRNA" space. The assumption of incompleteness of our data is quite realistic in the sense that in a microarray experiment, we usually do not observe protein concentrations together with mRNA concentrations due to the technical difficulty involved in performing such experiments. Thus we see the data as just noisy measurement of mRNA concentrations, whose dynamics can be described by some hidden process which involves protein transcription factors and mRNA concentrations. Another advantage of fitting SSM to the data stems from the fact that the variables of interest in the form of gene expression such as mRNA and protein transcription factors are seen as random variables, allowing the representation of some stochasticity, which could arise from either the measurement process or the nature of the biological process.

In microarray analysis, the number of predictors in the form of genes to be analyzed far exceeds the number of observations ($p >> n$). Faced with such explosion of data, regularization has become an important ingredient and is fundamental to high-dimensional statistical modeling. The Lasso (Tibshirani, 1996) is one of the few methods for shrinkage and selection in regression analysis that incorporates an L_1 regularization constraint to yield a sparse solution. A considerable amount of literature has been published on regularization methods in areas with large data sets such as genomics. These studies include the followings:

1. The regularization paths for the support-vector machine (Hastie et al., 2004).

2. The elastic net (Zou and Hastie, 2005) for applications with unknown groups of predictors and useful for situations where variables operate in correlated groups.

3. L_1 regularization paths for generalized linear models (Park and Hastie, 2007) and

4. The graphical lasso (Friedman et al., 2008b) for sparse covariance estimation and undirected graphs.

Gene regulatory networks are usually sparse. Also, molecular ontologies suggest few connections among the many thousands of genes i.e, each gene may only be regulated by a few number of other genes or transcriptions factors . For that reasons, we will expect many of the parameters to be zero leading to a sparse solution. It is in this context that we employ a regularization approach for the estimation of the parameters. This, form the basis for the L_1 penalization. The proposed method in the maximization step of the EM-algorithm is the L_1 penalty through a simple modification of the LARS algorithm by Efron et al. (2004), (Least Angle Regression). LARS is an efficient algorithm for computing the entire regularization path for the Lasso.

State space models are good robust candidates to represent interactions between biological components in the form of mRNA concentrations and protein transcription factors. We present a statistical method that infers the complexity, the dependence structure of the networks topology and the functional relationships between the genes, and deduce the kinetic structure of the networks. We estimate all model interaction parameters in order to clarify and describe the complex transcriptional response of a biological system and to clarify interactions between components. By so doing, we are able to add some useful interpretations to the model. We use the minimum AICc to determine the optimum level of sparsity.

The rest of this chapter is organized as follows. In section 4.2, we recall our genomic SSM introduced in chapter 2 , and give it a precise biological interpretation. Section 4.3 describes the inference method and the model selection technique. We perform a simulation study and "in silico" validation experiment in order to evaluate the performance of our method in section 4.4. Section 4.5 is the application of our model to a real data (T-cell data) and summary of our results. We conclude with a discussion of the method used, possible extension, and a summary of related work in section 4.6.

4.2 Genomic State Space Model

As previous our model is defined through the following dynamics.

Firstly the state dynamics or the state of the network satisfies an input dependent first-order Markov process

$$x_{tr} = Fx_{t-1,r} + Ay_{t-1,r} + \eta_{tr}. \tag{4.1}$$

where F is a regulatory matrix that quantifies the effect of the latent variables at consecutive time points and is of dimension $k \times k$. The quantity A represents the input-to-state matrix whose dimension is $k \times p$, $r = \{1, 2, ..., n_R\}$ denotes the number of biological replicates and η_{tr} is the Gaussian noise with mean 0 and variance-covariance matrix Q. The initial state x_0 is Gaussian distributed with mean $a_0 = 0$ and variance-covariance Q_0

Secondly the p observation dynamics y_t is a possibly time-dependent linear transformation of a k dimensional real-valued x_t with observational Gaussian noise ξ_t and is given by

$$y_{tr} = Zx_{t,r} + By_{t-1,r} + \xi_{tr}. \tag{4.2}$$

where Z describes how the latent variables in the form of transcription factors regulate the transcription of genes and is of dimension $k \times p$. The matrix B represents either degradation or production matrix of mRNAs also known as input-to-observation matrix whose dimension is $p \times p$ and ξ_{tr} is the measurement Gaussian noise with mean 0 and variance-covariance matrix R.

The framework captures the stochastic nature of our biological process and their dynamics. The model assumes that the evolution of the hidden variables x_t is governed by the state dynamics which follows an input dependent first-order Markov process. In essence we build a dynamic model that connects the observed variables y_t (RNA transcripts) to the k dimensional real valued unobserved quantities x_t such as unmeasured typically protein regulators.

The model indicates two space networks, the protein space and the mRNAs space, across consecutive time points. It assumes RNA-protein translation at two consecutive time points through the matrix A, and instantaneous protein-RNA transcription through Z.

A biological interpretation of the model network is also represented in Figure (4.1) which describes two fundamental stages in gene regulation which are in conformity with the central dogma which states that DNA does not code for protein directly but rather acts through 2 stages, namely, transcription and translation. The latent variables x in this model can be interpreted as TFs and interaction between them are modeled by Equation (4.1). We limit our work to linear interactions. A possible alternative is to assume nonlinear interactions but it substantially complicates the analysis. The input-to-state matrix A also known as observation-to-state matrix

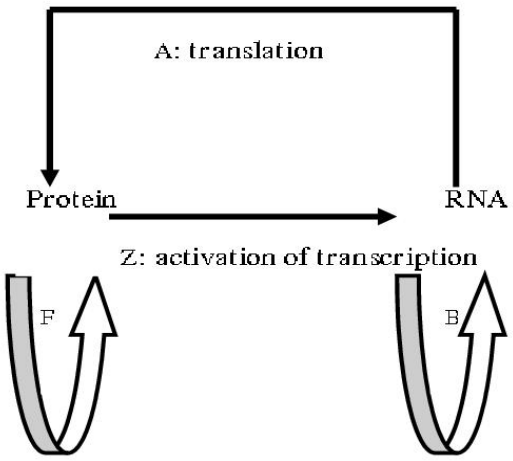

Figure 4.1. Biological interpretation of the input SSM.

models the influence or the effects of the gene expression values from previous time steps on the hidden states. The matrix B indicates the direct gene-gene interactions. The state dynamic matrix F describes the temporal development of the regulators or the evolution of the latent variables from previous time step $t-1$ on the current time step t. It provides key information on the influences of the hidden regulators on each other. As before we collect the model interaction parameters into a single vector φ i.e $\varphi = \{G, Q, R, Q_0\}$ where G is interpreted as a directed and weighted adjacency matrix of the graph of interactions.

4.3 Learning States and Parameters

4.3.1 The likelihood function

We now write the marginal likelihood function $l_y^m(\varphi)$ of the data y. This is given by

$$\begin{aligned} l_y^m(\varphi) &= \int \prod_{t=1}^{T} P(x_t|F, A, x_{t-1}, y_{t-1}) P(y_t|B, Z, x_t, y_{t-1}) dx \\ &= \int \prod_{t=1}^{T} \psi(x_t|\tilde{x}_t, \sigma_\eta^2 I) \psi(y_t|\tilde{y}_t, \sigma_\xi^2 I) dx. \end{aligned} \tag{4.3}$$

where $\psi(\mu, \Sigma)$ is the pdf of $N(\mu, \Sigma)$.

From now onwards for a given replicate and for simplicity, we will write the unpenalized complete log-likelihood as:

$$l_{y,x}(F, A, Z, B) = \sum_{r=1}^{n_R} l^r_{y_r x_r}(F, A) + \sum_{r=1}^{n_R} l^r_{y_r x_r}(Z, B) \tag{4.4}$$

Learning the parameters of a state space model including the hidden variable can be tackled from different approaches. Beal et al. (2005) inferred the parameters in the SSM using a Bayesian approach through a Variational Bayes Method (VBM) that approximates the posterior quantities required for Bayesian learning. As a probabilistic model, Wild et al. (2004) estimated the parameters through a frequentist approach using maximum likelihood inference in the context of EM algorithm. In our context, the number of parameters to be estimated $P = k^2 + 2kp + p^2$ far exceeds the number of observations. Thus we want to shrink unnecessary coefficients to zero. This will make interpretation of results easier and reflects the true underlying situation by introducing some level of sparsity. The formulation of our problem becomes:

$$\hat{\varphi} = \arg\max_{\varphi}(l^m_y) \tag{4.5}$$

subject to the constraints

$$||Z||_1 \leq s_1, \qquad ||B||_1 \leq s_2, \qquad ||A||_1 \leq s_3, \qquad ||F||_1 \leq s_4 \tag{4.6}$$

where s_i represents the regularization parameters or penalty parameters and we allow different penalty parameters for different coefficients. Equations (4.5) and (4.6) are called constrained regression problem viewed in this context as penalized state space models. Our L_1 constraint not only promotes sparsity but also minimizes the identifiability problems.

To find the solution to the above problem, many well developed procedures can be used. For example, quadratic programming (Tibshirani, 1996), the shooting algorithm (Fu, 1998), local quadratic approximation (Fan and Li, 2001) and most recently, the LARS method by Efron et al. (2004) can all be employed. Our proposed method adapt the later procedure; optimization under L_1 constraint, where a penalty term is added to the likelihood function giving rise to a penalized likelihood criterion. LARS or optimization with L1-regularization constraint turns out to be helpful and computationally feasible approach for finding sparse solutions in high dimension and by so rendering model interpretation easier.

4.3.2 The EM algorithm

As the true state variable is hidden, the integral in Equation (4.3) is difficult. We then applied the EM algorithm described in previous chapters with penalty constraint to obtain Penalized Maximum Likelihood Estimators (PMLEs) of φ using multivariate normal theory. The EM algorithm is an iterative method for finding the Maximum Likelihood Estimation (MLE) of φ using the observed data y_t, by successively maximizing the conditional expectation of the complete data likelihood given the observed values.

The expected log-likelihood function: The E-step.

Let $\mathbf{Q}$ denote the expected log-likelihood. Then from Equation (4.4), dropping the replicate index, $\mathbf{Q}$ becomes

$$\begin{aligned}
\mathbf{Q}(\varphi|\varphi^*) &= E_{x,\varphi^*}\left[l_{y_t,x_t}(\varphi)|\varphi^* y\right] \\
&= \sum_{t=1}^{T} E_{x,\varphi^*}\left[l_{y_t,x_t}(\varphi)|\varphi^* y\right] \\
&= \sum_{t=1}^{T} E_{x,\varphi^*}\left[l_{y_t}(Z,B)|y\right] + \sum_{t=1}^{T} E_{x,\varphi^*}\left[l_{x_t}(F,A)|y\right] \\
&= \mathbf{Q}_1(Z,B) + \mathbf{Q}_2(F,A) \qquad (4.7)
\end{aligned}$$

where $\varphi^* = (Z^*, B^*, F^*, A^*)$ is the estimate obtained from the previous M-step,

$$\begin{aligned}
\mathbf{Q}_1(Z,B) &= C_1 + 2\sum_{t=1}^{T} E(x_t^{'}) Z y_t + 2\sum_{t-1}^{T} y_{t-1}^{'} B^{'} y_t \\
&- \sum Z^{'} E(x_t^{'} x_t) Z - 2\sum_{t=1}^{T} E(x_t^{'}) Z B y_{t-1} \\
&- \sum_{t=1}^{T} B^{'} y_{t-1}^{'} y_{t-1} B \qquad (4.8)
\end{aligned}$$

and

$$\begin{aligned}\mathbf{Q}_2(F,A) &= C_2 + 2\sum_{t=1}^{T} y_{t-1}^{'} A^{'} E(x_t) + 2\sum_{t-1}^{T} E(x_t^{'}) F E(x_{t-1}) \\ &- \sum_{t=1}^{T} F^{'} E(x_{t-1}^{'} x_{t-1}) F - 2\sum_{t=1}^{T} y_{t-1}^{'} A^{'} F E(x_{t-1}) \\ &- \sum_{t=1}^{T} A^{'} y_{t-1}^{'} y_{t-1} A \end{aligned} \tag{4.9}$$

C_1 and C_2 are known constants. The first two moments needed in the E-step are supplied by the Kalman smoothing algorithm through a forward filtering pass and a backward smoothing pass. The above implies that for each replicate we run the Kalman smoothing algorithm to find the expected hidden states and their variance-covariance components and these are joined together to get $\mathbf{Q}(\varphi|\varphi^*)$. Now Equation (4.7) is the sum of two quadratic functions $\mathbf{Q}_1$ and $\mathbf{Q}_2$ that do not depend on x but rather depend on the parameters and the data y in a quadratic way. We maximize these two functions during the maximization step.

The update equations: The M-step.

At this stage we solve for

$$\hat{\varphi} = \arg\max_{\varphi} \mathbf{Q}(\varphi|\varphi^*) \tag{4.10}$$

subject to the constraints defined in Equation (4.6).

In essence we maximize iteratively the quadratic function $\mathbf{Q}$ given in Equation (4.7) across φ using LARS algorithm, where each coefficient is assigned a tuning parameter s. This breaks down to two maximization problems, one for $\mathbf{Q}_1$ across (Z, B) and the other for $\mathbf{Q}_2$ across (F, A). The iterative maximization process is similar in both cases.

We now show the maximization process for $\mathbf{Q}_1$. To do that, the following lemma is needed.

Lemma 4.3.1. *The solution that maximizes the quadratic function*

$$\mathbf{Q}(\mathcal{X}) = 2d^{'}\mathcal{X} - \mathcal{X}^{'} S \mathcal{X} \qquad \textit{subject to} \quad ||\mathcal{X}||_1 \quad \leq s \tag{4.11}$$

is given by the lasso solution

$$(\mathbf{y}-\mathbf{C}\beta)^{'}(\mathbf{y}-\mathbf{C}\beta) \qquad subject \quad to \quad ||\mathcal{X}||_1 \quad \leq s \tag{4.12}$$

where

$$\mathbf{C}^{'}\mathbf{C}=S, \qquad \beta=Vec(\mathcal{X}), \qquad \mathbf{y}=\mathbf{C}S^{-1}d. \tag{4.13}$$

and the LARS solution of (4.12) is a function of S *and* $d=\mathbf{C}'\mathbf{y}$

Proof. Properties of Gaussian distribution and Gaussian processes suggest that the quadratic $\mathbf{Q}(\mathcal{X})$ corresponds to a Gaussian $N(S^{-1}d, S^{-1})$. Therefore

$$\begin{aligned}
\mathbf{Q}(\beta) &= (\beta-S^{-1}d)^{'}\Sigma^{-1}(\beta-S^{-1}d) \\
&= (\beta-S^{-1}d)^{'}S(\beta-S^{-1}d) \\
&= (\beta-S^{-1}d)^{'}\mathbf{C}^{'}\mathbf{C}(\beta-S^{-1}d) \\
&= (\mathbf{C}S^{-1}d-\mathbf{C}\beta)^{'}(\mathbf{C}S^{-1}d-\mathbf{C}\beta) \\
&= (\mathbf{y}-\mathbf{C}\beta)^{'}(\mathbf{y}-\mathbf{C}\beta).
\end{aligned} \tag{4.14}$$

Suppose we have a set of linearly independent covariates $(\mathbf{c}_1, ..., \mathbf{c}_k)$ and we define the matrix

$$\mathbf{C}=(..., s_i\mathbf{c}_i, ...)_{i\in\mathbb{A}} \tag{4.15}$$

where $\mathbb{A}$ is the subset of the indices $\{1, ..., k\}$ of the active set and s_i are the signs and equal ± 1. Suppose

$$\mathbb{G}_{\mathbb{A}}=(\mathbf{C}_{\mathbb{A}}^{'}\mathbf{C}_{\mathbb{A}})^{-1} \quad \text{and} \quad A_{\mathbb{A}}=\left[1_{\mathbb{A}}^{'}\mathbb{G}_{\mathbb{A}}1_{\mathbb{A}}\right]^{-\frac{1}{2}} \tag{4.16}$$

where $1_{\mathbb{A}}$ is a vector of 1's of length equals the size of $\mathbb{A}$.

The unit vector making equal angles, less than 90^0, with the active columns of $\mathbf{C}_{\mathbb{A}}$ is given by

$$w_{\mathbb{A}}=A_{\mathbb{A}}\mathbb{G}_{\mathbb{A}} \tag{4.17}$$

Let

$$\hat{\gamma}=\underset{i\in\mathbb{A}^c}{min}^{+}\left\{\frac{(\hat{Q}_{max}-\hat{q}_i)}{A_{\mathbb{A}}-a_i}, \frac{(\hat{Q}_{max}+\hat{q}_i)}{A_{\mathbb{A}}+a_i}\right\} \tag{4.18}$$

where $\mathbf{q}=\mathbf{C}^{'}\mathbf{y}$, representing vector of current correlations, Q_{max} is the maximum absolute value from the set $\mathbf{q}$, $\mathbf{a}=(\mathbf{C}_{\mathbb{A}^c})^{'}\mathbf{C}_{\mathbb{A}}w_{\mathbb{A}}$ being the inner product vector.

Now the next step of LARS algorithm updates the coefficient $\hat{\beta}_{k-1}$, say to

$$\hat{\beta}_k = \hat{\beta}_{k-1} + \hat{\gamma} w_{\mathbb{A}} \tag{4.19}$$

We need to show that $\hat{\beta}_k$ is a function of $\mathbf{C}^{'}\mathbf{C}$ and $\mathbf{C}^{'}y$

The above definitions of $\mathbf{q}$ and $\mathbf{a}$ indicate that $\hat{\gamma}$ is a function of $\mathbf{C}^{'}\mathbf{C}$ and $\mathbf{C}^{'}y$. Therefore $\hat{\beta}_k$ is also a function of $\mathbf{C}^{'}\mathbf{C}$ and $\mathbf{C}^{'}y$ □

Now from Equation (4.8), given B we carry out the maximization process of $\mathbf{Q}_1$ across Z. Therefore, we can write $\mathbf{Q}_1(Z, B)$ as

$$\mathbf{Q}_1(Z) = c_1 + 2b_1^{'}Z - Z^{'}S_1Z \tag{4.20}$$

subject to

$$\sum_{j=1}^{kp} |z_j| \leq s_2 \tag{4.21}$$

where $S_1 = E(x_t^{'}x_t)$, b_1 is just a function of (y, B, x), and c_1 is a constant.

Applying lemma 4.3.1, the update maximum likelihood estimates $\hat{Z}$ from Equation (4.20) are just a function of S_1 and b_1. Therefore, we obtained the updates estimates $\hat{Z}$ by supplying the LARS function, the quantities S_1 and b_1 with a given tuning parameter where b_1 becomes the new data and S_1 the data matrix.

Next, given $\hat{Z}$, we maximize the quadratic function

$$\mathbf{Q}_1(B) = c_2 + 2b_2^{'}B - B^{'}S_2B \tag{4.22}$$

subject to

$$\sum_{j=1}^{p^2} |b_j| \leq s_1 \tag{4.23}$$

where $S_2 = \sum_{t=1}^{T} y_{t-1}^{'}y_{t-1}$, b_2 is just a function of (y, Z, x), c_2 is a constant. With the same analysis, the updates estimates $\hat{B}$ are obtained by supplying the LARS function, the quantities S_2 and b_2 with a given tuning parameter. Similar analysis is conducted for the estimation of F and A).

The advantage of this approach is that we see the LARS updates as functions not of the raw data, but instead as functions of S and b. This enables us to avoid first, the Cholesky decomposition of S and second, computing S^{-1} which are both time consuming and computationally inefficient.

1. Iterate across penalty parameters $s \in S$
   a. Start with initial values of φ
      i. Do the E-step by calculating the Kalman smoother
      ii. Perform the M-step via LARS algorithm
   b. Repeat (i) and (ii) until convergence
2. Across S select model with minimum AICc

Table I. Summary of the EM for Penalized Likelihood inference method

4.3.3 Model selection: Choice of regularization parameter s

Determining the optimal SSM tuning parameter s is an important issue. We apply Akaike's Information Criterion (AIC) method for our model selection. We generate a vector of values for the tuning parameters $s_{i,(i=1,\ldots,4)}$. For each combination of the values of the tuning parameters we run the EM algorithm and obtain

$$\hat{\varphi}(s) = \arg\max_{\varphi} [\mathbf{Q}(\varphi)]$$

subject to constraints in Equation(4.6).

$$AIC_c(\hat{\varphi}(s)) = -2l(y_t) + 2P\left[\frac{N}{N-P-1}\right] \tag{4.24}$$

where $N = pTn_R$ represents total number of observations and $P = p^2+2kp+k^2$ is the total number of estimated parameters. Then for each model, the AICc is computed and the model with the minimum AICc is selected. In essence, minimizing AICc, we obtained the optimal tuning parameter which is given by

$$\hat{s} = \arg\min_{s}[AICc(\hat{\varphi}(s))]$$

and the selected model parameters are given by $\hat{\varphi}(\hat{s})$. Table (I) summarizes the general formulation of the EM-L_1 penalized inference method:

4.4 Validation of Method

4.4.1 simulated data

In this section we evaluate our method on a simulated data based on the model described in Equations (4.1) and (4.2) with 10 different time points, $p = 3$ as number

Figure 4.2. The full true network G (left) and the full recovered network $\hat{G}$ (right)

Figure 4.3. Gene-gene interactions $ZA + B$. The true network(left) and the estimated network(right)

of genes and $k = 2$ hidden variables. In applying our algorithm, parameters were initialized as follows: Z and F are assumed to be identity matrices whiles we initialize A to be zero. For B we perform a simple linear regression where we regress current genes on its previous ones and R assumes the usual variance estimate from the regression.

The true (left) and the recovered (right) networks are depicted in Figure (4.2). The efficiency of our method is seen from the number of true links recovered. We also reported the gene-gene interactions matrix $\hat{Z}\hat{A} + \hat{B}$ in Figure (4.3). The matrix $\hat{Z}\hat{A} + \hat{B}$ captures all information related to gene-gene interactions for consecutive time points. It relevance stems from the fact it is identifiable.

Table II depicts the performance of our method as the network increases. For a fixed k, we increase p and monitor how best the network is recovered through TPR, FPR, and F_1 score. We experience a relatively stable F_1 score and TPR but a decrease in FPR as the number of nodes increase. To this, our method performs quite well even on a large network.

p	10	15	25	30
TPR	0.63	0.50	0.43	0.50
FPR	0.05	0.04	0.02	0.02
F-score	0.56	0.47	0.43	0.47

Table II. Simulation result for TPR, FPR and F-score as p, the number of nodes increase

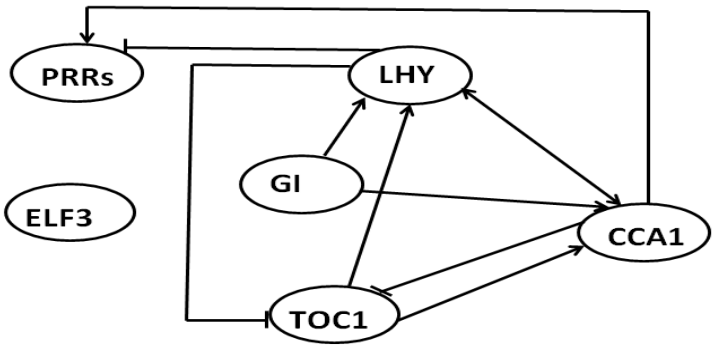

Figure 4.4. The Arabidopsis thaliana clock network.

4.4.2 In silico experiment: Arabidopsis thaliana clock

The question we address here is to check whether the links recovered by our method are actually reproducible. To do that we want to validate the model of the Arabidopsis thaliana oscillator by Salome and McClung (2004). The Arabidopsis is a model plant system that results from a combination of forward and reverse genetic approaches together with transcriptome-scale gene expression analyses. We consider a simple model of Arabidopsis clock made up of 9 genes namely, CCA1, LHY, TOC1, ELF4, ELF3, GI, PRR9, PRR5 PRR3 with 3 replicates. Most importantly Salome and McClung (2004) focus on the interaction between 4 out the 9 genes, namely LHY, CCA1, TOC1, and GI. The gene CCA1 has its corresponding protein named CIRCADIAN CLOCK ASSOCIATED 1. The gene LHY encodes a single Myb domain protein and is closely related to CCA1, both are important for proper clock function. Figure (4.4) depicts the genomic interaction in the Arabidopsis thaliana clock recovered by Salome and McClung (2004). It reveals a regulation activities

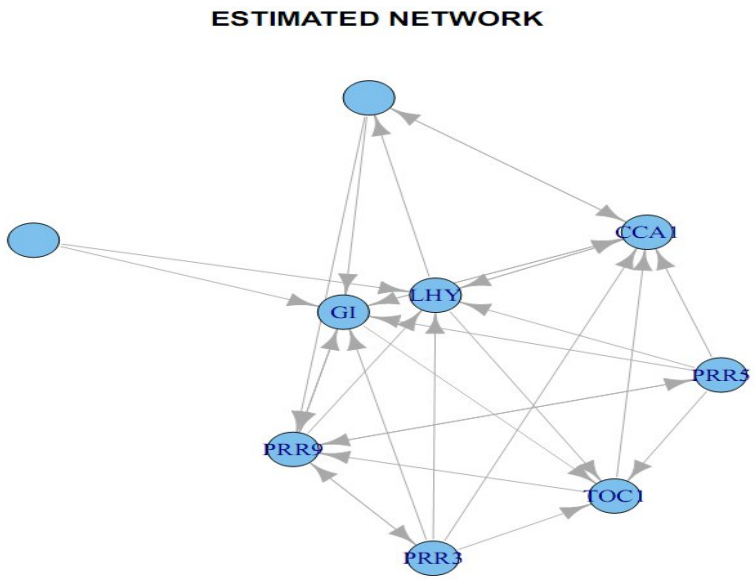

Figure 4.5. Reverse genetic appraoch on Arabidopsis thaliana clock. Nodes refers to genes expression whiles empty nodes indicates latent variables in the form TFs.

between CCA1 and LHY, this link came about with the analysis of the TOC1 promoter and closed the loop of the Arabidopsis clock. The model from Salome and McClung (2004) reveals that TOC1 acts as a positive regulator of the expression levels of CCA1 and LHY. The model also posits the repressive activity of CCA1 and LHY on TOC1. The gene GI is also necessary for high-level expression from the CCA1 and LHY.

In an attempt to recover the model of Arabidopsis thaliana clock by Salome and McClung (2004), we applied our method to the data with 2 hidden or latent variables in the form of transcription factors. Our recovered network is shown in Figure (4.5). For easy comparison purpose, we also focus on the interaction between two subnetworks representing the interaction between CCA1, TOC1, GI, LHY. The two subnetworks are subnetworks from Figure (4.4) and Figure (4.5). Both subnetworks supports the hypothesis that TOC1 is a positive regulator of the expression level of CCA1. Our result also confirms the interaction between CCA1 and LHY. However our method has failed to recover the negative regulatory activity of CCA1 and LHY on TOC1. Table (III) compares the two networks in terms of how many correct links we have recovered.

		True network		
		Links	No Links	Total
Estimated network	Links	6	2	8
	No Links	2	2	4
	Total	8	4	12

Table III. Comparison of a model of the Arabidopsis thaliana oscillator subnetwork (True subnetwork) and the subnetwork recovered by our method (Estimated subnetwork). The true subnetwork comprises of 8 links out of which our method recovered 6 correctly.

4.5 Application

Following from chapter (3), section (3.3.4), we have assumed the dimension of the hidden state k to be 4 and we applied our L_1 penalized inference method to the t-cell time course gene expression data introduced in section (3.5) of chapter (3). For each replicate, y_t and x_t consist of 45 genes and 4 transcriptions factors respectively, each, measured at 10 different time points, i.e for each replicate r, y_t and x_t are of dimension (45×10), (4×10) respectively. Some of these genes include RB1, CCNG1, TRAF5, CLU.... We estimated a total number of 2401 parameters consisting of B, A, Z and F. To do that, we iterate across the penalty parameters namely 4 different sequence of tuning parameters s_B, s_A, s_Z and s_F. While LARS produces the entire path of solutions, we make prediction or extract coefficients from the fitted LARS model using the `predict` function in LARS. The predict function allows one to extract a prediction at a particular point along the path. This procedure is repeated until convergence. We then have different set of estimated model parameters corresponding to each set of tuning parameters. At this stage, we applied model selection technique via minimum AICc described in section 4.3.3 to select the optimum parameters. At the end, we obtained the connectivity matrix of the directed genomic graph. The estimated optimum tuning parameters has given rise to fairly sparse networks.

The outputs are graph showing connections from one gene expression variable at a given time point t to another gene expression variable whose expression it influences at the next time point, $t + 1$. The output depicted in Figure (4.6) is a sub-network that shows the topology of gene FYB. We found that the genes such as CCNA2, FYB, and CASP8 are mostly activation genes. Specifically, FYB activates the expression level of genes such as GATA3, CCNA2, CD69, IL3RA while CASP8 activates genes

such as: JUND, CDC2, CD69. Figure (4.7) recovers the interactions between the Jun proteins family and other genes. It identifies JUND to have significant number of connections in the form of activation and inhibitions. The structure of the network is visualized using the R package for Network analysis and visualization `igraph`.

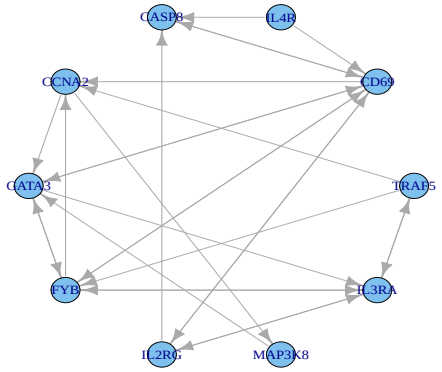

Figure 4.6. Subnetwork found representing the topology of gene FYB in connection with some selected genes

Our method has resulted in relatively sparse networks as compared to the approach in chapter (3). In all, the following genes were found to have the highest number of interactions in terms of inwards directed connections: TRAF5, C3X1, CASP4, CDK4 and IL3RA. In addition, from a topological point of view genes such as JUND, AKT1, FYB, and CCNA2 occupy a crucial position in the recovered networks. We recommend these genes to be object of further study by biologist. These results support the works of Rangel et al. (2004); Wild et al. (2004). Both found gene FYB to occupy an important position in their respective graphs. At the optimum turning parameter, we found JUNB interacting directly with CASP4 through JUND; a result also supported by Beal et al. (2005). The unpenalized inference approach from chapter (3) has indicated that JUNB activates directly CDC2. This work also supports the same interaction. A portion of the sub-network found by Beal et al. (2005) and Andrea et al. (2010) representing the interactions between CASP4 and JUND is also found in our network through Figure (4.7). Another interesting interactions that was supported by previous literature were interactions between JUND and CASP7 on one hand and interactions between JUND and CDC2 both in the

form of inhibitions. JUND is predicted to repress the expression level of the cell cycle regulator CDC2. This clearly supports the hypothesis that JUND negatively regulates cell growth and acts as anti-proliferative and anti-apoptotic signal genes.

Our work reveals that genes such as AKT1 and MLC1 also occupy a crucial position. AKT1 is found to influence the expression level of many genes. Of these, include the JUN proteins JUNB , one interleukin receptor gene IL3RA, one apoptosis-related cysteine protease CASP4, the cell division cycle CDC2. AKT1 is also seen to activate the expression level of a transcription factor. In our model, MLC1 is found to regulate positively the expression level of one of the transcription factors. Also MLC1 represses several genes including CASP7, CDK4, C3X1 to mention but few. Andrea et al. (2010) found that CAPS4 inhibits the expression level of CAPS8. This interaction was supported by our work. Beal et al. (2005) did not recover such interaction. Also some findings, through Figure (4.7) of our current study do not support the work of Beal et al. (2005) in the sense that we found no interactions between JUND and Caspase-7 and also between JUNB and MAPK8. Thus, the results based on our methodology suggest some findings that are supported by the current literature and are biologically interpretable, while some other findings have not been documented yet in biological literature and we hope these new findings will be confirmed in the near future.

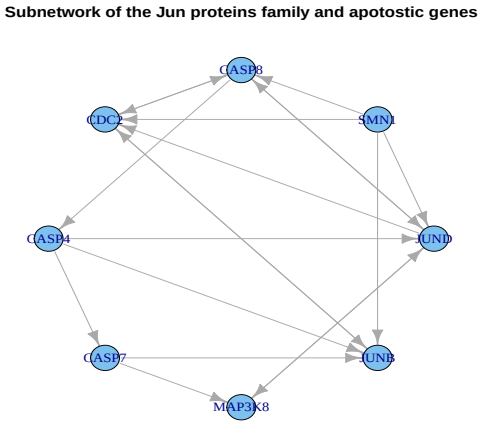

Figure 4.7. Subnetwork found representing the interactions between Jun proteins family and other genes

4.6 Conclusion

In this chapter, we have inferred a sparse dynamic network by using an input dependent penalized linear state space model. We have assumed that the true biological process is not fully observed and the hidden variables were first calculated using a Kalman filtering and smoothing algorithm via an E-step. We then proceed to update the model parameters through an L_1 regularization constraint via LARS algorithm in the maximization step. We used AICc to determine the optimum combination of tuning parameters and hence the model parameters.

The method we have presented in this chapter can be viewed as an Expectation-Regularization-Maximization approach which produces sparse high-dimensional gene dynamic regulatory networks. Most importantly, the LARS algorithm adopted guarantees us interpretable model, and accurate predictors. The advantages of using linear SSM stems from the fact that the linearity assumption has resulted in a more stable network and has enabled us to recover the dynamics of the network easily as compared to the nonlinear relationships. The inference method via LARS is potentially revolutionary, offering interpretable models, relative stability, accurate predictions, unbiased inferences, and a nice graphical display of coefficient paths that indicates the key tradeoff in model complexity. We used the R package for Network analysis and visualization `igraph` to display simple, and easy to understand graph through which the whole system under study can be ascertained quite easily.

Gene regulatory interactions surely include complex interactions which nonlinear SSM may capture well. To recover networks from nonlinear models is however complicated in both theory and computationally, especially in high dimensional setting. Future works will encompass extending the linear SSM into a Non Linear State Space Model whose hidden process will be defined through an integration of an Ordinary Differential Equation and estimate both parameters and hidden variables through the same inference technique. We also plan to overcome the ODE limitations namely ability to handle noisy data and the high number of model parameters by integrating a sparse ODE model into a graphical model framework, thus taking noisy measurement into account, and the resulting model will then be embedded into a penalized maximum likelihood learning set-up.

Appendix A

R-package: glassomix

A.1 Descrption

The software `glassomix` provides a general framework for networks recovering through a model-based soft clustering. It provides functions for parameter estimation via the Penalized EM algorithm for Gaussian graphical mixture models in high dimensional setting. The main function is `glasso.mix` upon which a model selection is performed. The package estimates the optimum number of mixture components K and the tuning parameter (lambda) based on the Extended Bayesian Information Criteria -EBIC- via `select.gm` function. The graphical structural of the K networks are also plotted through the function `gm.plot`

A.2 `glassomix-package`

A package for high dimensional Undirected Graphical Mixture Models selection. This package provides an implementation of the procedures described in chapter (2). The main function is `glasso.mix`. This function performs the graph estimation using glasso and a model selction is performed based on Extended Bayesian Information Criterion through the function `select.gm`. The graphical structural of the K subgroups of population of individuals is estimated and plotted via the function `gm.plot`.

A.3 The functions

A.3.1 `glasso.mix`: sparse Gaussian undirected graphical mixture model estimation

This is the main function that performs the inference via EM algorithm. This function for each value of K, estimates the responsibility matrices ($n \times K$) at the E-step and then given these probabilities, estimates the precision matrices at the M-step via glasso.

Usage

```
glasso.mix(data,K=NULL,lambda=NULL,em.iter,n.lambda,
penalize.diagonal=TRUE, ebic.gamma=0.5,Kmax)
```

Arguments

- `data`: $n \times p$; rows = n, number of observation, columns = p, number of graph nodes/variables.
- K: A sequence of integers denoting the number of mixture components (clusters).
- `lambda`: (Non-negative) regularization parameter for glasso. lambda=0 means no regularization. It could be a scalar or a vector.
- `em.iter`: The maximum number of EM iteration.
- `n.lambda`: The length of the tuning parameter lambda.
- `penalize.diagonal`: Should diagonal of precision matrix be penalized? Default is FALSE.
- `ebic.gamma`: The Extended Bayesian Information Criteria parameter, usually ebic.gamma is between 0 and 1.
- `Kmax`: The maximum number of K.

Value

The details of the output components are as follows:

1. res: A list with the following components:

- `loglik`: A vector value of un-penalized log-likelihood for each value of K.
- `naiveloglik`: A vector value of naive log-likelihood extracted from glasso for each value of K.
- `n.par`: Total number of estimated parameters in each of the precision matrices corresponding to each value of K at the various regularization parameters.
- `bestlambda.ebic`: Optimal tuning parameter corresponding to K.
- `besttheta.ebic`: The penalized precision matrix corresponding to the optimal EBIC for each value of K.
- `bestpi.ebic`: The mixture proportion corresponding to the optimal EBIC for each value of K.
- `Theta.Pen`: Penalized precision matrices corresponding to each value of K at the various regularization parameters.
- `Theta.NonPen`: Non-penalized precision matrices corresponding to each value of K at the various regularization parameters.
- `pi.ind`: Responsibility matrices $(n \times K)$ corresponding to each value of K for the various regularization parameters. It can also be seen as vector of probabilities $(w_{i1}, ..., w_{iK})$ of individual i belonging to the k classes at penalty λ.
- `pi`: K Mixing coefficients for the various regularization parameters.
- `EBIC`: All EBIC values for each value of K at the various regularization parameters.

2. `lambda`: The sequence of regularization parameters used.
3. `Kmax`: The maximum number of mixture components.
4. `n.lambda`: The length of the tuning parameter lambda.
5. `data`: The data matrix.

A.3.2 `gm.plot`: Graphical plot of the K networks

Plots the optimum K precision matrices corresponding to the optimum EBIC.

Usage

```
gm.plot(output)
```

Arguments

`output`: It is a list which is the result of `select.gm` function. It shows the graphical representation (dependencies) of the p-variables in each cluster.

A.3.3 `select.gm`: High dimensional sparse Gaussian graphical mixture model selection

This function selects the optimal model according to Extended Bayesian Information Criterion (EBIC) for EM- algorithm for parameterized High dimensional sparse Gaussian graphical mixture models. The function estimates the optimum number of mixture components and the regularization parameter lambda.

Usage

```
select.gm(ret)
```

Arguments

`ret`: It is a list which is the result of `glasso.mix` function. It Implements the model selection clustering through a model selection based on the EBIC for a parametrized Gaussian graphical mixture model across K for each of the regularization parameters.

Value

The details of the output components are as follows:

- `n.cluster`: Optimal number of clusters or mixture components.
- `eBIC`: All EBIC values.
- `lambda.eBIC`: Optimum lambda value based on minimum EBIC.
- `Th.Pen`: $n.cluster$ precision matrices.
- `Th.NPen`:$n.cluster$ non-penalized precision matrices.
- `Pi.ind`: Optimum responsibility matrices ($n \times n.cluster$) corresponding to the soft-K-means clustering.
- `Pi`: Optimum mixture proportions based on EBIC criterion.

- clusters: $(n \times 1)$ vector containing the indices of the clusters where the data points are assigned to.

- `Pen.LogLik` : The un-penalized log-likelihood corresponding to the optimal EBIC.

- `NPen.LogLik`: The naive un-penalized loglikelihood corresponding to the optimal EBIC.

- `lambda`: The sequence of regularization parameters used.

A.3.4 `summary.glasso.mix`: Summary of results

Reduced summary of the result according to `glasso.mix`.

Usage

```
summary(object,...)
```

Arguments

object: an object with S3 class `glasso.mix`. A list of the result from the function `glasso.mix`.

value

The details of the output components are as follows:

- `lambda`: The sequence of regularization parameters.

- `pi`: Mixture proportions for each K across lambda.

- `bestlambda.ebic`: Optimum lambda value based on EBIC for each K.

- `besttheta.ebic`: The penalized precision matrix corresponding to the optimal EBIC for each value of K.

- `n.par`: Total number of estimated parameters in each of the precision matrices corresponding to each value of K at the various regularization parameters.

A.3.5 summary.select.gm: Summary according to the model selection function `select.gm`

Usage

```
summary(object,...)
```

Arguments

object: an object with S3 class `select.gm`. A list of the result from the function `select.gm` function.

value

The details of the output components are as follows:

- `mix.comp`: Optimal number of clusters.
- `lambda.eBIC`: Optimum lambda value based on EBIC.
- `clustering`: $(n \times 1)$ vector containing the indices of the clusters where the data points are assigned to.
- `mix.prop`: Optimum mixture proportions.

Appendix B

Proof of lemma 2.2.1

We give a proof of the existence and uniqueness of the strongly consistent MLE.

Let $f_\gamma(\mathbf{y})$ be a probability density function of a vector variable $\mathbf{y} \in R^n$ and a vector parameter $\gamma \in R^v$. If $\{y_i\}_{i=1}^n$ is an independent sample of observations on a random variable $\mathbf{y} \in R^n$ whose probability density function is $f_{\gamma_0}(\mathbf{y})$ for some $\gamma_0 \in R^v$, then an MLE of γ_0 is a choice of γ which locally maximizes the log likelihood function $l_\mathbf{y}(\gamma)$.

If f is a differentiable function of γ, a necessary condition for an MLE is that, for the likelihood equations,

$$\frac{\partial l}{\partial \gamma_l} = 0; \quad l = 1, ..., \upsilon$$

be satisfied, where γ_l is the l^{th} component of γ. In the following, the objective is to show that, if f satisfies certain conditions, then given any sufficiently small neighborhood of γ_0, there is, with probability, 1 as the sample size n approaches infinity, a unique solution of the likelihood equation in that neighborhood, and this solution is a MLE of γ_0.

Assume that $f_\gamma(\mathbf{y})$ satisfies assumptions A1-A2: Let

$$H(\gamma) = \begin{pmatrix} \frac{1}{n}\frac{\partial l}{\partial \gamma_1} \\ \vdots \\ \frac{1}{n}\frac{\partial l}{\partial \gamma_\upsilon} \end{pmatrix}$$

Clearly if the likelihood equations are satisfied, $H(\gamma) = 0$ and by the law of large numbers, $H(\gamma_0)$ converges to 0 with probability, 1. Next, it follows from conditions A1-A2 that there exists a neighborhood ρ_0 of γ_0 (contained in ρ and, for convenience, convex) and a positive ϵ such that, with probability, 1 as n approaches infinity,

$\nabla H(\gamma) < -\epsilon I$ for all $\gamma \in \rho_0$,where ∇H denotes the vector of partial derivatives of H with respect to the coordinates of γ . Denoting the spherical neighborhood of radius δ about γ_0 by ρ_δ, we establish the following:

Lemma B.0.1. *With probability* 1 *as* n *approaches infinity*

1. H *is one to one on* ρ_0,
2. $H(\rho_\delta)$ *contains the ball of radius* $\epsilon\delta$ *about* (γ_0) *whenever* $\rho_\delta \subseteq \rho_0$

Proof. We may assume that $\nabla H(\gamma) < -\epsilon I$ for all $\gamma \in \rho_0$, since the probability that this is the case is 1 as n approaches infinity. To prove lemma B.0.1 (1), suppose that $H(\gamma_1) = H(\gamma_2)$ for γ_1 and $\gamma_2 \in \rho_0$. Then

$$\begin{aligned} 0 &= (\gamma_1 - \gamma_2)' [H(\gamma_1) - H(\gamma_2)] \\ &= (\gamma_1 - \gamma_2)' \left\{ \int_0^1 \nabla H [\gamma_2 + t(\gamma_1 - \gamma_2)] \, dt \right\} (\gamma_1 - \gamma_2) \end{aligned} \tag{B.1}$$

The negative-definite aspect of ∇H implies that $\gamma_1 = \gamma_2$, and lemma B.0.1 (1) is proved. □

To prove lemma B.0.1 (2), suppose that $\rho_\delta \subseteq \rho_0$, and let γ_1 be a boundary point of ρ_δ. Then,

$$H(\gamma_1) - H(\gamma_0) = \left\{ \int_0^1 \nabla H [\gamma_0 + t(\gamma_1 - \gamma_0)] \, dt \right\} (\gamma_1 - \gamma_0)$$

After left-multiplying this equation by $(\gamma_1 - \gamma)'$, one verifies using Schwarz's inequality and the negative-definite aspect of ∇H that

$$||H(\gamma_1) - H(\gamma_0)|| > \epsilon ||\gamma_1 - \gamma_0|| = \epsilon\rho$$

where $||\quad||$ denotes the usual Euclidean norm on R^v. Since all boundary points of $H(\rho_\delta)$ are images under H of boundary points of ρ_δ, the proof of lemma B.0.1 (2) is complete.

Bibliography

Agakov, Felix. V., Orchard, Peter. and Storkey, Amos. J. (2012), Discriminative mixtures of sparse latent fields for risk management., *Journal of Machine Learning Research - Proceedings Track* **22**, pp. 10–18.

Akaike, H. (1974), A new look at the statistical model identification, *Automatic Control, IEEE Transactions on* **19**(6), pp. 716 – 723.

Andrea, Rau., Jean-Louis, Foulley., Florence, Jaffrezic. and Rebecca, W.Doerge. (2010), An empirical bayesian method for estimating biological networks from temporal microarray data, *Statistical Applications in Genetics and Molecular Biology* **9, Issue 1 2010**.

Banfield, Jeffrey. D. and Raftery, Adrian. E. (1993), Model-based gaussian and non-gaussian clustering, *Biometrics* **49**(3), pp. pp. 803–821.

Beal, M. J., Falciani, F., Ghahramani, Z., Rangel, C. and Wild, D. L. (2005), A bayesian approach to reconstructing genetic regulatory networks with hidden factors, *Bioinformatics* **21**, pp. 349–356.

Bernardo, JM (2003), Bayesian clustering with variable and transformation selections, *Bayesian Statistics 7: Proceedings of the Seventh Valencia International Meeting*, Oxford University Press, USA, p. 249.

Bernhard, O. Palsson (2011), *Systems Biology Simulation of Dynamics Network States*.

Biernacki, Christophe., Celeux, Gilles. and Govaert, Gérard (2000), Assessing a mixture model for clustering with the integrated completed likelihood, *IEEE Trans. Pattern Anal. Mach. Intell.* **22**(7), pp. 719–725.

Bower, M. and Bolouri, H. (2001), *Computational Modeling of Genetic and Biochemical Networks*.

Bozdogan, Hamparsum (1983), Testing the number of components in a normal mixture.

Bremer, M. and Doerge, R. W. (2009), The km-algorithm identifies regulated genes in time series expression data., *Advances in Bioinformatics* .

Brown, R. G. and Hwang, P. Y. (1997), Introduction to random signals and applied kalman filtering, *John Willey and Sons, New York* .

Buntine, Wray (1995), Chain graphs for learning, *In Uncertainty in Artificial Intelligence*, Morgan Kaufmann, pp. 46–54.

Burnham, K. P. and Anderson, D. R. (2002), *Model Selection and Multi-Model Inference*, Vol. 2.

Cao, Jiguo. and Zhao, Hongyu. (2008), Estimating dynamic models for gene regulation networks, *Bioinformatics* **24**(14), pp. 1619–1624.

Chanda, K. C. (1954), A note on the consistency and maxima of the roots of likelihood equations, *Biometrika* **41**(1/2), pp. pp. 56–61.

Chen, Kuang-Chi., Wang, Tse-Yi., Tseng, Huei-Hun., Huang, Chi-Ying. F. and Kao, Cheng-Yan. (2005), A stochastic differential equation model for quantifying transcriptional regulatory network in saccharomyces cerevisiae, *Bioinformatics* **21**(12), pp. 2883–2890.

Cramer, H. (1946), *Mathematical methods of statistics*, Princeton University Press.

Day, N. E. (1969), Estimating the components of a mixture of normal distributions, *Biometrika* **56**(3), pp. pp. 463–474.

de la Fuente, Alberto., Bing, Nan., Hoeschele, Ina. and Mendes, Pedro. (2004), Discovery of meaningful associations in genomic data using partial correlation coefficients, *Bioinformatics* **20**(18), pp. 3565–3574.

Dempster, A. P., Laird, N. M. and Rubin, D. B. (1977), Maximum likelihood from incomplete data via the em algorithm, *Journal of the Royal Statistical Society. Series B (Methodological)* **39**(1), pp. pp. 1–38.
URL: *http://www.jstor.org/stable/2984875*

Dent, W. and Min, A.S. (1978), A monte carlo study of autoregressive integrated-moving average processes, *Journal of Econometrics* **7**, pp. 23–55.

Derisi, J. L., Iyer, V. R. and Brown, P. O. (1997), Exploring the metabolic and genetic control of gene expression on a genomic scale, *Science* **278**, pp. 680.

Dewey, T. G. and Galas, D. J. (2000), Generalized dynamical models of gene expression and gene classification, *Funt. Int. Genomics* **1**, pp. 269–278.

Efron, B. (1979), Bootstrap methods: Another look at the jacknife, *The Annals of Statistics* **7**, pp. 1–26.

Efron, B., Hastie, T., Johnstone, I. and Tibshirani, R (2004), Least angle regression, *Annals of statistics* **32**, pp. 407–451.

Fahrmeir, L. and Kunstler, R. (2009), Penalized likelihood smoothing in robust state space models, *Biometrika (1999)* **49**, pp. 173–191.

Fahrmeir, L. and Wagenpfeil, S. (1997), Penalized likelihood estimation and iterative kalman smoothing for non-gaussian dynamic regression models, *Computational Statistics & Data analysis (1997)* **24**, pp. 295–320.

Fan, J. and Li, R. (2001), Variable selection via nonconcave penalized likelihood and its oracle properties, *Journal of the American Statistical Association* **96**(456).

Fang-Xiang, Wu., Wen-Jun, Zhang. and Anthony, J. Kusalik. (2004), Modelling gene expression from microarray expression data with state-space equations., *Biocomputing* **9**, pp. 588–592.

Friedman, Jerome., Hastie, Trevor. and Tibshirani, Robert. (2008a), Sparse inverse covariance estimation with the graphical lasso, *Biostatistics* **9**(3), pp. 432–441.

Friedman, Nir. (2004), Inferring cellular networks using probabilistic graphical models, *Science* **303**(5659), pp. 799–805.

Friedman, T., Hastie, T. and Tibshirani, R. (2008b), Sparse inverse covariance estimation with the graphical lasso, *Biostatistics* **9**, pp. 432–441.

Fu, Wenjiang. J. (1998), Penalized regressions: The bridge versus the lasso, *Journal of Computational and Graphical Statistics* **7**(3), pp. pp. 397–416.

Fujita, Andre., Sato, Joao., Garay-Malpartida, Humberto., Yamaguchi, Rui., Miyano, Satoru., Sogayar, Mari. and Ferreira, Carlos. (2007), Modeling gene expression regulatory networks with the sparse vector autoregressive model, *BMC Systems Biology* **1**(1), pp. 39.

George, Casella and Berger, R. L. (1996), *Statistical Inference*.

Ghahramani, Z. and Hinton, GE. (1996), Parameter estimation for linear dynamical system., *Technical report, University of Toronto* .

Hastie, S. Rosset., Tibshirani, R. and Zhu, J. (2004), The entire regularization for the support vector machine., *Journal of Machine Learning Research* **8**, pp. 1519–1555.

Horn, R. A. and Johnson, C. R. (1990), Matrix analysis.

Husmeier, Dirk. (2003), Sensitivity and specificity of inferring genetic regulatory interactions from microarray experiments with dynamic bayesian networks, *Bioinformatics* **19**(17), pp. 2271–2282.

Jensen, F. (1946), *An introduction to Bayesian networks*, Vol. 74, UCL Press London.

Kauffman, S.A. (1993), *The origins of order*, Vol. 22:437.

Kiefer, J. and Wolfowitz, J. (1956), Consistency of the maximum likelihood estimator in the presence of infinitely many incidental parameters, *The Annals of Mathematical Statistics* **27**(4), pp. pp. 887–906.

Kishino, H. and Waddell, P. (2000), Correspondence analysis of genes and tissue types and finding genetic links from microarray data, *Genome Informatics* **11**, pp. 83–95.

Lauritzen, Steffen, L. (1996), *Graphical models*, Vol. volume 17 of Oxford Statistical Science Series.

Lauritzen, Steffen, L. and Wermuth, N. (1989), Graphical models for associations between variables, some of which are qualitative and some quantitative, *The Annals of Statistics* **17**(1), pp. pp. 31–57.

Ljung, L. and Caines, P.E (1979), Asymptotic normality of prediction error estimators for approximate systems models, *Stochastics* **3**, pp. 29–46.

Lo, Yungtai., Mendell, Nancy. R. and Rubin, Donald. B. (2001), Determining the number of component clusters in the standard multivariate normal mixture model using model-selection criteria., *Biometrika* **88**(3), pp. 767–778.

McLachlan, G. J., Bean, R. W. and Peel, D. (2002), A mixture model-based approach to the clustering of microarray expression data, *Bioinformatics* **18**(3), pp. 413–422.

Meinhold, R. J. and Singpurwalla, N. D. (1983), Understanding the kalman filter, *The Americam Statistician* **37, N0. 2**, pp. 123–127.

Nicolai, Meinshausen., Peter, Bhlmann. and Eth, Zrich. (2006), High dimensional graphs and variable selection with the lasso, *Annals of Statistics* **34**, pp. 1436–1462.

Nir, Friedman., Michal, Linial. and Iftach, Nachman. (2000), Using bayesian networks to analyze expression data, *Journal of Computational Biology* **7**, pp. 601–620.

Pan, Wei. and Shen, Xiaotong. (2007), Penalized model-based clustering with application to variable selection, *J. Mach. Learn. Res.* **8**, pp. 1145–1164.

Park, M. Y. and Hastie, T. (2007), l_1 regularization path algorithm for generalized linear models., *Journal of the Royal Statistical Society* **Series B,69**, pp. 659–677.

Patrik, Dhaeseleer., Liang, Shoudan. and Somogyi, Roland. (2000), Genetic network inference: from co-expression clustering to reverse engineering, *Bioinformatics* **16**(8), pp. 707–726.

Pearl, J. (2000), *Causality:models, reasoning, and inference*.

Peer, Dana, Regev, Aviv, Elidan, Gal and Friedman, Nir (2001), Inferring subnetworks from perturbed expression profiles, *Bioinformatics* **17**(suppl 1), pp. S215–S224.

Perrin, B.E., Ralaivola, L., Mazurie, A., Bottani, S., Mallet, J. and d'Alche Buc, F. (2003), Gene networks inference using dynamic bayesian networks., *Bioinformatics 19 (Suppl.2)* pp. 138–148.

Quach, Minh., Brunel, Nicolas. and Florence, d'Alch-Buc. (2007), Estimating parameters and hidden variables in non-linear state-space models based on odes for biological networks inference, *Bioinformatics* **23**(23), pp. 3209–3216.

Ramsay, J. O., Hooker, G., Campbell, D. and Cao, J. (2007), Parameter estimation for differential equations: a generalized smoothing approach, *Journal of the Royal Statistical Society: Series B (Statistical Methodology)* **69**(5), pp. 741–796.
URL: *http://dx.doi.org/10.1111/j.1467-9868.2007.00610.x*

Rangel, C., Angus, J., Ghahramani, Z., Lioumi, M., Sotheran, E., Gaiba, A., David, L. Wild. and Falciani, F. (2004), Modeling t-cell activation using gene expression profiling and state-space models, *Bioinformatics* **20**(9), pp. 1361–1372.

Redner, R.A. (1980), *Maximum Likelihood Estimation for Mixture Models*, JSC (Series), Lyndon B. Johnson Space Center, NASA.

Redner, Richard (1981), Note on the consistency of the maximum likelihood estimate for nonidentifiable distributions, *The Annals of Statistics* **9**(1), pp. pp. 225–228.

Roweis, Sam and Ghahramani, Zoubin (1999), A unifying review of linear gaussian models, *Neural Comput.* **11**(2), pp. 305–345.

Ruan, Lingyan, Yuan, Ming and Zou, Hui (2011), Regularized parameter estimation in high-dimensional gaussian mixture models, *Neural Comput.* **23**(6), pp. 1605–1622.

Sachs, Karen., Perez, Omar., Pe'er, Dana., Lauffenburger, Douglas. A. and Nolan, Garry. P. (2005), Causal protein-signaling networks derived from multiparameter single-cell data, *Science* **308**(5721), pp. 523–529.

Salome, Patrice. A. and McClung, Robertson. C. (2004), The arabidopsis thaliana clock., *Journal of Biological Rhythms* .

Schfer, Juliane. and Strimmer, Korbinian. (2005), An empirical bayes approach to inferring large-scale gene association networks, *Bioinformatics* **21**(6), pp. 754–764.

Schwarz, G. (1978), Estimating the dimension of a model, *The Annals of Statistics* **6(2)**, pp. 461–464.

Shumway, R. H. and Stoffer, D. S. (2005), *Time series analysis and its applications with R examples*, Vol. second edition.

Shumway, R.H. (2000), Dynamic mixed models for irregularly observed time series, *Resenhas-Reviews of the Institute of Mathematics and Statistics,University of Sao Paulo,USP Press, Brazil* **4,No.4**, pp. 433–456.

Shumway, R.H. and Stoffer, D.S. (1982), An approach to time series smoothing and forecasting using the em algorithm, *J.Time series Analysis* **3**, pp. 253–264.

Surajit, R. and Lindsay, Bruce. G. (2005), The topography of multivariate normal mixtures, *Annals of Statistics* **33**(5), pp. pp. 2042–2065.

Tibshirani, R. (1996), Regression shrinkage and selection via the lasso., *Journal of the Royal Statistics Society* **series B, 58**, pp. 267–288.

Wald (1949), Note on the consistency of the maximum likelihood estimate, *The Annals of Mathematical Statistics* **20**(4), pp. pp. 595–601.

Wen, Xiling., Fuhrman, Stefanie., Michaels, George, S., Carr, Daniel, B., Smith, Susan., Barker, Jeffery, L. and Somogyi, Roland. (1998), Large-scale temporal gene expression mapping of central nervous system development, *Proceedings of the National Academy of Sciences* **95**(1), pp. 334–339.

Whittaker, Joe. (2009), *Graphical Models in Applied Multivariate Statistics*, Wiley Publishing.

Wild, D. L., Rangel, C., Angus, J. and Ghahramani, Z. (2004), Modeling genetic regulatory networks using gene expression profiling and state space models, *Probabilistic Modelling in Bioinformatics and Medical informatics. Springer-Verlag* .

Wille, A. and Peter, Buhlmann. (2006), Low-order conditional independence graphs for inferring genetic networks, *Statistical Applications in Genetics and Molecular Biology* **5**(1).

Xintao, Wu., Yong, Ye. and Kalpathi, R. Subramanian (2003), Interactive analysis of gene interactions using graphical gaussian model, *ACM SIGKDD Workshop on Data Mining in Bioinformatics, 3:6369.*

Yamaguchi, R. and Higuchi, T. (2006), State-space approach with the maximum likelihood principle to identify the system generating time-course gene expression data of yeast, *Int. J. Data Mining and Bioinformatics* **1, N0. 1**, pp. 77–87.

Yamaguchi, R., Ryo, Y., Seiya, I., Tomoyuki, H. and Satoru, M. (2007), Finding module-based gene networks with state-space models, *IEEE Signal Processing Magazine[37]* .

Yuan, Ming. and Lin, Yi. (2007), Model selection and estimation in the gaussian graphical model, *Biometrika* **94**(1), pp. 19–35.

Zhou, Hui., Pan, Wei. and Shen, Xiaotong. (2009), Penalized model-based clustering with unconstrained covariance matrices, *Electron J Sta* **3**, pp. 14731496.

Zou, H. and Hastie, T. (2005), Regularization and variable selection via the elastic net, *Journal of the Royal Statistical Society* **Series B, 67(2)**, pp. 301–320.

Zoubin, Ghahramani. (2001), Introduction to hidden markov models and bayesian networks, *International Journal of Pattern Recognition and Artificial Intelligence* **15(1)**, pp. 9–42.

Summary

This thesis is concerned with the problem of genomic networks modelling and inference from non-homogenous data or from high-throughput data sources such as microarray gene expression data. We have derived a statistical method that infers the complexity and the conditional independence structure of the network topology, the functional relationships between the genes and deduce the kinetic structure of the network. The diverse nature of biological systems will increasingly require modelling tools that can be used to properly design and interpret biological networks. We have proposed two novel networks inference methods, namely penalized Gaussian graphical mixture model and a penalized state space model.

Gaussian graphical models explore dependency relationships between random variables by estimating the corresponding precision matrix. We have considered the problem of large networks reconstruction from heterogeneous data using a Gaussian graphical mixture model (GGMM). Parameters were learned through a penalized maximum likelihood technique by imposing an L_1 penalty constraint on the precision matrix. We have provided general consistency results for the penalized maximum likelihood estimator in the Gaussian graphical mixture model. Our results indicate a better performance in parameter consistency as well as in graph selection consistency with penalty rates λ_n proportional to $n^{-1/2}$. Our method is suitable for recovering large networks from non-homogeneous data.

Next, to incorporate a temporal dependency structure in the model and to account for latent or hidden variables in the process we introduced a state space model. Inferring dynamic networks involving hidden or latent variable is an important challenge. Using a dynamic state space representation, we devised a method for inferring regulatory networks from high-dimensional data using linear Gaussian state space models. Our method is based on an EM algorithm with an incorporated Kalman smoothing algorithm in the E-step to calculate the hidden states. We obtained an explicit formulation of the parameters defining our state space model. Parametric bootstrap was used to determine the selection of parameters and edge selection or deletion is done through hypothesis testing at a particular significance level α. We

have used the AIC to determine the hidden state's optimal dimension.

Because of the high-dimensional nature of such data coupled with a sparsity assumption, the maximum likelihood approach for parameter learning is prone to be noisy. Also, parameter identifiability can be an issue. A natural way to avoid the above problems is through regularization. We have built an input dependent SSM and employed a penalized maximum likelihood inference in the context of the EM algorithm. This is achieved via a modified version of the LARS algorithm. As a result, we are able to add some useful biological interpretation to the obtained estimates.

Samenvatting

Dit proefschrift gaat over het modelleren en de inferentiële analyse van genetische netwerken met behulp van niet-homogene data van typisch automatisch meetmethodes, zoals gen expressie microarray technieken. Wij presenteren een statistische methode die de complexiteit en de afhankelijkheidsstructuur van een netwerktopologie, die de functionele relaties tussen genen weergeeft, kan vaststellen en we leiden de kinetische structuur van dit netwerk af. De vraag naar gepast gereedschap om biologische netwerken te ontwerpen en te interpreteren neemt toe. En door de ontwikkeling hiervan kan een steeds grotere verscheidenheid aan biologische systemen bestudeerd worden. Wij stellen twee nieuwe methoden voor netwerkinferentie voor: gepenaliseerde Gaussische grafische samengestelde modellen and *state-space* modellen.

Met behulp van Gaussische grafische modellen kunnen afhankelijkheidsrelaties tussen stochasten onderzocht worden door de bijbehorende precisiematrix te schatten. Het probleem van netwerkenreconstructie op basis van heterogene data hebben we aangepakt door een Gaussische grafisch samengesteld model (GGSM) te introduceren. De parameters in dit model worden geschat met behulp van een gepenaliseerde maximum likelihood techniek, waarbij een L_1-penalisatie op de precisiematrix wordt uitgevoerd. Dit is een belangrijke vernieuwing op dit gebied; we formuleren het Gaussische samengestelde modelleerprobleem in de context van Gaussisch grafisch modelleren. Het GGSM is in staat clusters van individuele componenten te identificeren. We bewijzen ook de consistentie van het gepenalizeerde meest waarschijnlijke schatter.

Om rekening te houden met tijdsafhankelijkheid en latente variabelen hebben we het state-space model (SSM) geïntroduceerd. Het analyseren van dynamische netwerken, waarin latente variabelen een rol spelen, is een uitdaging. Door gebruik te maken van een dynamische state-space representatie in de vorm van lineaire Gaussische state space modellen, hebben we een methode ontworpen om netwerken te bepalen op basis van hoog-dimensionale data. De schatttingsmethode hierbij is gebaseerd op een EM-algoritm met een *Kalman smoother* algoritme in de E-stap om de latente toestand te berekenen. We verkrijgen vervolgens een expliciete formulering

van de parameters, die ons state space model definiëren. Tot slot gebruiken we parametrische *bootstrap* om de parameters te selecteren en voegen relaties in het grafische model in of verwijderen deze op basis van het testen van hypotheses bij een bepaald significantie niveau α. We gebruiken de het AIC criteria om het optimale aantal latente variabelen te bepalen.

Vanwege het hoog-dimensionale karakter van de data die we bestuderen, is de maximum likelihood benadering gevoelig voor ruis. Bovendien, de identificeerbaarheid van de parameters is in deze situatie ook niet vanzelfsprekend. Een natuurlijke manier om deze problemen te vermijden is door middel van regularisatie. We definiëren daarom een input-afhankelijk SSM en gebruiken een gepenaliseerde maximum likelihood strategie in de context van het EM algoritme. Hierbij maken we gebruik van een aangepaste versie van het LARS algoritme. Tot slot presenteren we nuttige interpretaties van het model in specifieke gevallen.

Acknowledgments

I would like to express my special appreciation and thanks to my supervisor Prof. Ernst Wit for accepting me as a PhD student at the beginning of 2010. He guided my first steps into state space modeling and later introduced me into penalized graphical models for network inference.

I am also indebted to the member of the reading of committee Prof. Edwin van den Heuvel, Prof. Mark van de Wiel, and Dr. Veronica Vinciotti for their helpful comments.

A warm thanks go to all my office-mates and colleagues, whom I was extremely lucky to have met. A special appreciation goes to Antonino for his kind support at the beginning of my PhD, followed by Ivan, Nynke, Reza and Parya, Javier and Fentaw for having made themselves available in supporting me in many ways.

I am also indebted to the personnel of the department and the office of the Graduate school of Science for their enormous support. In particular I wish to extend my sincere appreciation to Ineke Schelhaas, Esmee Elshof, Desiree Hansee to mention but a few for their dedication and hard work.

I would like to express my sincere gratitude to my family and to all the people I have met throughout my PhD. They have all contributed in one way or another to the great experience I have had. For limited space, I may not be able to list all their names and I hope they will forgive me.

Groningen, 2014 Anani. Lotsi.